Macintosh Forensics

A Guide for the Forensically Sound Examination of a Macintosh Computer

Ryan R. Kubasiak, Investigator - New York State Police

About The Author 5

Contact Information 6

Email 6

Telephone 6

Fax 6

Mail 6

About This Document 7

Tools Needed and Requirements of the Document 8

Digital Examination Overview 9

File System 12

Operating Systems 12

Data Files 12

MAN Pages 13

Technologies 14

Bonjour 14

FileVault 14

Spotlight 15

UNIX and the FreeBSD System 15

Microsoft Windows on a Mac? 15

Disk Arbitration 16

Activate/Deactivate Disk Arbitration 16

Target Disk Mode 18

Target Disk Mode Procedure 19

The Macintosh Boot Process 20

Open Firmware and Extensible Firmware Interface 20

Startup Manager 20

BootX, boot.efi, and System Initialization 21

Boot EFI Utilities 23

rEFIt 23

Booting a Macintosh from the LiveCD 24

Imaging a Target Macintosh 25

Target Disk Mode 25

LiveCD 26

Drive Removal 26

Disk Structure 27

Apple Partition Map 27

GUID Partition Table 29

Journaling 31

FileVault and MacOS X Security 32

FileVault Preference Pane 32

sparseimage and User Home Directory 35

Acquire the Encrypted User Home Directory 36

DiskUtility and DMG Files 41

DiskUtility Features 41

DMG vs. sparseimage 41

Encrypted vs. Unencrypted 42

DD and Raw Images 42

Spotlight 43

User Home Directory Structure 44

User Library Folder - In Depth 46

Applications 48

Address Book 48

iCal 48

Mail 48

.Mac and Related Evidence 49

.Mac 49

Safari, and Other Web Browsers 50

Safari 50

iChat, and Instant Messaging Applications 51

iChat 51

Mac OS X Log Files 52

Mac OS X “plist” Files 53

Sleep and Safe Sleep 56

Detailed Macintosh Techniques 57

Apple Boot Key Combos 57

Create a Brute Force Dictionary File 58

Useful Artifacts and Commands 59

References 61

Websites 62

Recommended Utilities and Applications 63

MacOS X 10.4 Command Line Utilities and Daemons 64

About The Author

I began my foray into the world of computers in 7th grade. Our school laboratory was using Commodore 64 computers and the BASIC programming language. Soon, my parents purchased an Apple IIc for our home and I continued writing in BASIC, and now "Apple Logo" as well. My intrigue continued thru high school developing my skills in BASIC and the Pascal programming languages. Ultimately, I achieved Advanced Placement in Computer Science my senior year, yielding college credits.

I went on to the State University of New York at Buffalo and earned a Bachelor of Science in Computer Science and a Concentration in Mathematics. All of my schooling was done on the Macintosh LC, VAX/VMS and Sun Solaris based systems. We utilized Modula-2 and C as programming languages. C++ just wasn't prevalent enough during my college years. One of my favorite achievements of college was writing from scratch, an Assembly language code compiler. I also wrote a multi-tasking operating system for a fictitious Robot, and a dating service front and back end for a fictitious customer.

During school, and immediately after graduation, I worked for SUNY at Buffalo in the LAN Systems group. I went from a student assistant to full time employee and totaled 4 1/2 years with the university as an employee. As a LAN Administrator, I was charged with the setup, maintenance and upgrading of 4 public computing laboratories with hundreds of PC and Macintosh computers, and many office "node" sites with multiple PC and Macintosh computers. Along with the desktops, I also was charged with the operation and maintenance of Novell Netware and Microsoft NT based servers. I was a part of the team that also setup and maintained Remote Access Services and Tape Backups. The experience was invaluable towards the world of forensics, but didn't begin to educate me on the intricacies of a forensic examination.

I moved on from SUNY at Buffalo in 1998 to become a New York State Trooper. After 5 years "on the road", I was selected as a new member to the Computer Crime Unit. I have received two certifications, Encase Certified Examiner and Certified Computer Examiner. I hold multiple certificates from classes I have completed and have "expert witness" status in the criminal court system.

My passion in computing has always been the Apple platform. Starting with my first computer, the Apple IIc, I have owned a Mac LC, LCIII, Centris 650, PowerMac 8500AV, iMac G4 and Macbook Pro. I have been installing and configuring the operating system since "System 6" and I maintain a membership with the Apple Developer Network. I routinely following the development of the operating system itself with great interest.

I continue today to enhance my education, training, and investigative skills. My goal is to share some of what I have learned within this writing.

Contact Information

This is the first of what I hope to be many iterations of MacOS X information for the forensic investigator. In order to keep this relevant, I look forward to hearing from anyone and everyone! Here are a few of the ways you can get in touch with me:

Email

rkubasia@troopers.state.ny.us

Telephone

518-457-5712

Fax

518-485-5280

Mail

1220 Washington Avenue, Building 30

Albany, NY 12226

About This Document

This document is to guide a digital forensic examination of a Macintosh computer in the simplest yet sound manner. In order to accomplish this writing, you will notice a rather extensive bibliography. There are many great resources on the internet, in the local bookstore and via training sessions. It seems there is no "one" resource that begins to consolidate this information to create a reference. It is highly recommend that as a digital forensic examiner, you take advantage of the most current information available, utilizing this document and the sources cited. The most informative site on MacOS specifics will always be Apple Inc. You will see throughout that specific Apple Document reference numbers have been included for both credibility as well as future use when new technologies replace what is written here. Apple Inc. does not delete the texts posted on their site. Use this site and others for independent sources of what you plan to testify to.

There will never exist a complete guide to a forensic examination of any platform. There are near infinite directions a case may lead, as well as the fact that technology changes as quick as this document is being written. The goal of this first writing is to get solid, sound practices out to the Macintosh forensic community, and to follow up with additional documents that continue with these techniques, and include more in-depth looks at technologies not able to be noted here.

Images in this document are either created via screen capture on a live MacOS X system, or via the trademarked icons thru Apple Inc. All mentions of companies and their technologies are copyright/trademark of the respective entity.

References from the Apple Inc. Developer website or Support website are noted at the begining of the appropriate section. The document number is supplied for direct reference to the original writing.

No part of this document may be reproduced or utilized in any way without the express written permission of the author.

Tools Needed and Requirements of the Document

Target machine is assumed to be a Macintosh!

This guide is going to cover three different techniques to forensically "look" at the data of the target machine. Two techniques will involve directly using the target machine itself, while one will use another machine attached. To achieve all three of these forensic examinations, you will need to have with you:

- Macintosh OS X based laptop for mobile forensics, preferably an Intel for greater flexibility.
- Macintosh OS X based desktop for laboratory forensics, preferably an Intel system
- MacOS X 10.4 (or current) with the XCode tools installed
- LiveCD for both PowerPC and Intel
- Firewire cable with appropriate adapters
- USB Flash Drive, minimum of 1GB in size (4GB for creating a bootable USB drive)
- Examination Notes information sheet

This document will focus on OS X, heavily on version 10.4. Other versions will be mentioned and noted throughout.

Digital Examination Overview

Although crimes themselves have not changed, the methodology of committing them is ever changing. Our challenge is to keep pace with the digital aspect to all crimes. Investigations now must include a digital aspect as well as the traditional methods. Crimes of all levels are being plotted, planned or perpetrated with computers, PDAs, cell phones, USB flash drives, wrist watches, electronic pens, and others. The examiner needs to be cognizant of this, and trained to recognize these items. Specialized Examiners need to be continually educated and trained on current forensic techniques to analyze the data on these high tech devices. It simply is not acceptable to turn on a computer and see what is there!

First Responders are critical in initial actions taken such as on-site viewing of evidence and/or the securing of digital evidence. For this person, a checklist is not acceptable. An understanding of what needs to be done so one can adapt to the unique situations that present themselves is necessary. A loss of data or worse, corruption of data, at this point can severely jeopardize any case or situation.

Employers need to understand the importance of training, certification, and court presentation. A well qualified examiner, whether a First Responder or Specialized Examiner, will constantly stay up to date in technology advancements and training. For law enforcement, the National White Collar Crime Center offers excellent courses for the perfect price, free. There are many other options for training, most of which will be a financial *investment*. "Investment" is stressed because taking a course once is not good enough. Repeated training on newly emerging technology will be a must. Multiple colleges and universities have recognized and developed digital forensic classes, as well as degree programs. Also, software companies such as Black Bag Technologies, Guidance Software, and Access Data offer classes that concentrate on their specific software, yet teach useful skills in analysis. Courses and certifications that are publicly available vs. law enforcement only classes are preferred. Techniques that can reproduced by the digital forensic community at large are more revered in a courtroom setting.

The conditions, in criminal circumstances, to consider a limited scope examination rather than utilize a full laboratory analysis are:

- Facilitate Arrest - You have a search warrant and need to find evidence at the crime scene to facilitate and arrest of the target.
- Consent Search - You don't have anything more than permission from the target to look, but the permission is the look on-premises only.
- Exigent circumstances such as a missing person.

Field forensics is NEVER a substitute for a full-fledged, digital forensic laboratory. Working in an open environment such as a target's home or office presents dangers as well as opportunity for missed information. With that in mind, this guide is designed to safely and soundly guide the First Responder or Specialized Examiner to the data in a quick and forensically sound manner.

Three techniques are available to examine the target Macintosh. First, the Macintosh desktop/laptop/server can be booted into "single-user" mode. This state, as describe in-depth later, is a forensically sound state and allows for information to be gathered. In single-user mode, however, a thorough working knowledge of UNIX will be needed. Second, the same target machine can be booted from a LiveCD, such as MacOS X boot disk, a Knoppix distribution or Ubuntu LiveCD, and view the contents of the hard drive from it. Third, the target computer can be booted into Firewire Disk Mode (Target Disk Mode) and viewed from a secondary computer. Each of these techniques have benefits as well as pitfalls.

Single-User Mode utilizes an already installed operating system, features established by Apple, and greatest speed of previewing data. It also is command line driven, very much a manual process for setup, and potentially has been shut off or maliciously altered. Using the suspect's own operating system is almost always a bad idea, leading to potentially mistaken results.

LiveCD offers a known boot media with a known operating system each and every time you conduct a preview. It offers a well-known, always available set of tools for each and every limited scope examination conducted. It also is RAM intensive, will not always work with the latest hardware, or may not boot at all. Blackbag Technologies offers a subscription for a forensically sound Macintosh boot disk. It is also possible to create your own bootable disk that is both forensically sound and has specific utilities installed. The downside to creating your own disk is the lack of support for future machines. Apple Inc. does tweak the operating to take advantage of newer hardware. The specific changes from Apple come on a DVD with the specific computer. For instance, as of this writing, the MacOS X 10.4 box set available for purchase is for PowerPC Macintoshes only and will NOT boot Intel based systems.

Target Disk Mode offers the greatest flexibility. You are able to use your laptop (or desktop) with choice of operating system to look at the target machine. It yields the greatest speed and the widest variety of tools for examination. It also may not function at all on the target computer. This technology is discussed further, later in the document.

Every digital examination should involve the following steps:

- Physically secure evidence or conduct on-site preview (Collection)
- Acquisition of digital media
- Verification of acquired data
- Archive of acquired data with verification
- Analysis of acquired data
- Reporting of results

Only the first two allow for the usage of original evidence. Special care is taken during these steps to insure original evidence is not altered. This document is written entirely based on that care. If you do not wander outside of the scope of this document, you will not be altering original evidence. All techniques outside of this document should be well tested in a controlled environment for forensic soundness before attempting use on evidence.

A on-site examination typically will yield only a fraction of the evidence on a target computer. It may yield 0% evidence. It is NOT a substitute for a full, in-laboratory analysis. Just because it was not found during a limited scope examination, doesn't mean it's not there.

Macintosh Aspects

Apple has always been a very unique company, hence the operating system, file systems, and applications are also unique. Some basics to know and understand before looking at a Macintosh include the following:

File System

HFS+ (and the older HFS) are the two predominant file systems found on any Macintosh. Without "something" to recognize this file system, you will be left looking at a seemingly unallocated drive with raw data only. Tools such as Encase from Guidance Software and BBT Forensic Suite from BlackBag Technologies can appropriately interpret this file system and display the contents in a user friendly way. Also, the Macintosh itself knows how to display its own file system, and we use this fact when using Single-User mode, LiveCD, or the target disk mode.

A Macintosh may contain other file systems, just as any other computer. With the release of "Boot Camp" from Apple, Intel based systems could very well have NTFS, FAT32, EXT3, etc. The Intel based Macintosh computers are very capable of running multiple operating systems with multiple file systems. Always be aware of this when using techniques, and be aware of consequences.

Operating Systems

MacOS X and MacOS 9 are the two dominant operating systems that will be found on any Macintosh. With the release of "Boot Camp" from Apple, any operating system that operates on Intel hardware can be successfully installed and run. Just because an "Apple Logo" is displayed on the side of the computer doesn't mean an Apple operating system with be used. Apple has released Windows XP Service Pack 2 drivers as well as Windows Vista drivers, so expect those more often. Many hack websites have figured out how to use Boot Camp to install other operating systems and successfully boot. Just as common will be virtualization software such as Parallels, VMWare or VirtualPC. With these, you will encounter a "file" that actually contains an entire hard drive worth of data from a different operating system.

With that said, an extremely high percentage of Macs will be running OS X or OS 9. This document's focus will mostly be on the OS X based machines. OS X based PowerPC Macintoshes have the possibility of containing OS 9 "within" the OS X installation. It is referred to as "Classic" and is run simultaneously to the OS X environment.

Data Files

The Macintosh has used for several years, two "forks" to any file. They are the Resource fork and the Data fork. Apple has recommended to developers to discontinue the use of the Resource fork. If a Macintosh file is copied to a File System that doesn't support Resource forks, the fork will be lost. As an examiner, this is extremely important to know. If a file with a Resource fork is copied

to a Fat32 volume, for instance, the MacOS will handle the resource fork and open the file appropriately. However, the way in which it is handled is thru a hidden file. With an example file named "test.txt", one will notice a hidden file in the same directory named "._test.txt". This is the resource fork. MacOS X will copy his file from FAT32 correctly when the "test.txt" file is copied. Moving over to an operating system that doesn't recognize this, such as Microsoft Windows, the same copy will lose the Resource fork data. Resource forks can best be equated to Alternate Data Streams in the NTFS world.

Macintosh application files (or .app files) are actually not a single file at all. They are a folder, that is displayed via the Finder as single custom icon, and appropriately launched. If you Control-Click on an application file, you will notice the choice to "Show Package Contents". This will actually open the folder rather than launch the application. The contents have a small chance of being evidentiary in value, but the user data associated with an application is typically in the Home directory. Any folder can be made into an application by simply adding the ".app" extension to the name. However, when you double-click a self-made application, the Finder will likely give an error message because the application is not truly an application yet. Since an application is really just a specialized folder, problems occur if it is copied to a File System and opened within another operating system. Viewing MyApplication.app in a Windows environment will show a folder with the name of MyApplication.app. Further, the folder will open in windows and the Package contents will be seen, much like the "Show Package Contents" command.

Some applications actually use this package concept to create the data file. iWork has two applications, Keynote and Pages. They each save files in a Package format, and not a single flat file. Looking a MyDocument.pages on a FAT32 volume through Microsoft Windows will again result in a folder with the name MyDocument.pages and the folder will open when double-clicked. Be aware of this operation, and expect it when sharing files between operating systems.

Even more importantly, if you are examining a MacOS based system with a Windows tool, you WILL see package files differently than the intended view AND functionality. Certain portions of a forensic examination of a MacOS based system will require a Macintosh. Plan accordingly!

MAN Pages

One of the BEST features of each MacOS X based system is the help available. Specifically, the MAN pages available are perfect support documentation for any case. When you use a command line function, consider making the MAN page for that command a part of your report. The MAN pages are updated as system updates come out, making the output of the MAN page on the day of usage important. An easy way to do this is an output redirect. For example, if you are about to use the 'dd' command line utility, output the MAN page to a text file.

```
man dd > DD_MANPages.txt
```

This will output the MAN page entry to a text file. Save this text file in your case notes area for future reference. The best reference material an investigator can have is the materials supplied by the company itself!

Technologies

Mac OS X has some very robust technologies behind the Graphical User Interface. The operating system is UNIX derived, which gives us the power and support of a huge online community. The operating system has both a GUI and command line available. Within the OS, Applescript and shell scripting can be done, both allowing for the automation of processes and tasks.

Bonjour

Bonjour, formerly Rendezvous, is a technology developed by Apple to make network configuration and setup seamless to the end-user. Defined by Apple:

> Bonjour, also known as zero-configuration networking, enables automatic discovery of computers, devices, and services on IP networks. Bonjour uses industry standard IP protocols to allow devices to automatically discover each other without the need to enter IP addresses or configure DNS servers.

Bonjour is installed by default on OS X based machines running 10.3 or later. It is also available for download for Windows 2000 or XP based computers also.

FileVault

AES-128 encryption. FileVault automatically encrypts and decrypts the contents of your home directory on the fly. FileVault is off by default after initial setup or installation, but can be easily enabled. More about this technology later in the document.

Spotlight

Spotlight is the indexing engine and search technology used to keep track of files and their meta-data. A hidden file is created called "`.Spotlight-V100`" and contains the indexing data. Spotlight is enabled by default, and is not easily turned off for the entire system. More on Spotlight later in this document.

UNIX and the FreeBSD System

MacOS X, all versions, utilize the UNIX subsystem. This means, that for the first time, the MacOS is not only a GUI based system, but also is command line driven. This brings immense power and flexibility, along with the time tested stability of UNIX to the operating system. When researching How-To's on the MacOS X system, you can usually include generic UNIX information, as well as Linux equivalents. Many times, a Linux source code will be able to compile on the Macintosh with little changes.

Microsoft Windows on a Mac?

Yes. If the Macintosh is an Intel based system, a beta of the software called "Boot Camp" may be installed and Microsoft Windows XP SP2 or Vista may be installed. In addition, on both PowerPC and Intel based Macintoshes, emulation and virtualization software can be run allowing for other operating systems to run. Microsoft VirtualPC (formerly Connectix) is for the PowerPC based systems. The software was recently discontinued (but can still be purchased) because PowerPC Macintoshes have been discontinued. Newer software for the Intel Macintoshes such as SWSoft's Parallels Desktop or VMWare Fusion can run multiple, concurrent virtualized operating systems. These technologies will be discussed further.

Disk Arbitration

Disk Arbitration is a daemon in OS X that mounts file systems. This is the feature that automatically mounts and displays your USB Flash drive when you plug it in for instance. Disk Arbitration will mount volumes read/write, which is bad in the forensic world. When utilizing an OS X based Macintosh to preview another computer, Disk Arbitration needs to be "off".

Activate/Deactivate Disk Arbitration

1. Make a backup of the file "/etc/mach_init.d/diskarbitrationd.plist"
   a. "sudo cp /etc/mach_init.d/diskarbitrationd.plist /Backup/"
2. Remove /etc/mach_init.d/diskarbitrationd.plist
   a. "sudo rm /etc/mach_init.d/diskarbitrationd.plist"
   b. I HOPE YOU MADE THE BACKUP!
3. Reboot your system and Disk Arbitration is now off.
4. To turn Disk Arbitration back on, copy the original file back to its original location
   a. "sudo cp /Backup/diskarbitrationd.plist /etc/mach_init.d/"
5. Reboot your system and Disk Arbitration is now on.

As stated directly from the MAN pages:

DISKARBITRATIOND(8) BSD System Manager's Manual DISKARBITRATIOND(8)

NAME

diskarbitrationd -- disk arbitration daemon

SYNOPSIS

diskarbitrationd [-d]

DESCRIPTION

diskarbitrationd listens for connections from clients, notifies clients of the appearance of disks and filesystems, and governs the mounting of filesystems and the claiming of disks amongst clients.

diskarbitrationd is accessed via the Disk Arbitration framework.

Options:

-d Report detailed information in /var/log/diskarbitrationd.log. This option forces diskarbitrationd to run in the foreground.

The file /etc/fstab is consulted for user-defined mount points, indexed by filesystem, in the mount point determination for a filesystem. Each filesystem can be identified by its UUID or by its label, using the constructs ``UUID'' or ``LABEL'', respectively. For example:

```
UUID=DF000C7E-AE0C-3B15-B730-DFD2EF15CB91 /export ufs   ro
UUID=FAB060E9-79F7-33FF-BE85-E1D3ABD3EDEA none    hfs   rw,noauto
LABEL=The\040Volume\040Name\040Is\040This none    msdos ro
```

FILES

/etc/fstab
/etc/mach_init.d/diskarbitrationd.plist
/var/log/diskarbitrationd.log
/var/run/diskarbitrationd.pid

SEE ALSO

fstab(5)

Darwin July 18, 2004 Darwin

Results from a preview or analysis are only useful if everything has been conducted under forensically sound procedures. We must insure that everything done from start to finish guarantees unaltered data OR in a worst case scenario, results that are documentable, known changes to the target machine. We will NOT be purposefully trying to achieve the latter! The known changes and documentation should only be for a procedure attempted that did not result in the desired outcome. For instance, if you attempt to boot a target machine with a LiveCD and instead, the Mac OS boots, you must document what happened.

Target Disk Mode

(Apple Document 58583)

Target Disk Mode is a technology that allows a Macintosh computer to act as an external, firewire disk. The computer will not access the file system or other data in this state until user interaction causes this. Its an extremely useful tool for us. A separate note from Apple on this states:

> Tip: FireWire Target Disk Mode works on internal ATA drives only. Target Disk Mode only connects to the master ATA drive on the Ultra ATA bus. It will not connect to Slave ATA, ATAPI or SCSI drives.

This means we cannot access multiple installed drives with this method. If you know there are 2 or more drives in the target computer, consider the LiveCD method.

> In addition, the following models support the use of Target Disk Mode:
>
> iMac (Slot Loading) with Firmware version 2.4 or later
>
> iMac (Summer 2000) and all models introduced after July 2000
>
> eMac (all models)
>
> Mac mini (all models)
>
> Power Mac G4 (AGP Graphics) with ATA drive
>
> Power Mac G4 Cube
>
> Power Mac G4 (Gigabit Ethernet) and all models introduced after July 2000
>
> Power Mac G5 (all models)
>
> iBook (FireWire) and all models introduced after September 2000
>
> MacBook (all models)
>
> PowerBook G3 (FireWire)
>
> PowerBook G4 (all models)
>
> MacBook Pro (all models)

Target Disk Mode Procedure

To use Target Disk Mode in a forensically sound manner, use the following steps:

1. Make sure that the target computer is turned off. If you are using a laptop as the target computer, you should also plug in its AC power adapter.
2. Boot the target computer while holding down the **Option** key. This will yield one of two results. Either you will see a list of bootable devices (partitions) or you will see a prompt to enter the Firmware password. If the latter occurs, you **CANNOT** use Target Disk Mode.
3. Use a FireWire cable to connect the target computer to your computer. The forensic Macintosh (your computer) does not need to be turned off.
4. Start up the target computer and immediately press and hold down the T key until the FireWire icon appears. The hard disk of the target computer should become available to the host computer and will likely appear on desktop.
5. When you are finished with the examination, drag the target computer's hard disk icon to the Trash or select Put Away from the File menu (Mac OS 9) or Eject from the File menu (Mac OS X).
6. Press the target computer's power button to turn it off.
7. Unplug the FireWire cable.

To remain forensically sound, the Macintosh being used to view the Target should have DiskArbitration turned OFF.

If your examination machine is Windows based, be VERY cognizant of the possible writes being made to any FAT or NTFS partitions. The firewire connection is not write blocked in any way. For this reason, it is not recommended to use Target Disk Mode with a Windows based computer.

The Macintosh Boot Process

[The following section has wording taken verbatim from the Apple Developer website]

Open Firmware and Extensible Firmware Interface

Open Firmware and Extensible Firmware Interface are similar to the function of BIOS and are used on PowerPC and Intel based Macintoshes respectively.

When the power to a Macintosh computer is turned on, the BootROM firmware is activated. BootROM (which is part of the computer's hardware) has two primary responsibilities: it initializes system hardware and it selects an operating system to run. BootROM has two components to help it carry out these functions:

- POST (Power-On Self Test) initializes some hardware interfaces and verifies that sufficient memory is available and in a good state.
- On PowerPC-based Macintosh computers, Open Firmware initializes the rest of the hardware, builds the initial device tree (a hierarchical representation of devices associated with the computer), and selects the operating system to use. On Intel-based Macintosh computers, EFI does basic hardware initialization and selects which operating system to use.

If multiple installations of Mac OS X are available, BootROM chooses the one that was last selected by the Startup Disk System Preference. The user can override this choice by holding down the Option key while the computer boots, which causes Open Firmware or EFI to display a screen for choosing the boot volume.

> **Note:** On some legacy hardware, the same version of BootROM can start either Mac OS 9 or Mac OS X. Most current hardware can start only Mac OS X.

Startup Manager

(Apple Document 106178)

Startup Manager was introduced with these Apple computers and is present on these and all later models (including all Intel-based Macs):

- iMac (Slot Loading)
- iBook
- PowerBook (FireWire)
- Power Mac G4 (AGP Graphics)
- Power Mac G4 Cube

BootX, boot.efi, and System Initialization

[The following section is taken verbatim from the Apple Developer website]

Once BootROM is finished and a Mac OS X partition has been selected, control passes to the BootX (PowerPC) or boot.efi (Intel) boot loader. The principal job of this boot loader is to load the kernel environment. As it does this, the boot loader draws the "booting" image on the screen.

BootX and boot.efi can be found in the /System/Library/CoreServices directory on the root partition. In addition, a copy of boot.efi can be found at /usr/standalone/i386/boot.efi.

In "exotic" boot situations such as booting from a UFS volume, a software RAID volume, and so on, a copy of the boot loader is stored on a separate HFS+ "helper" volume to get the system started. In some versions of Mac OS X, a copy of the kernel and mkext cache are also included on the helper volume. In these cases, the booter and other components on the root volume are unused.

The boot loader first attempts to load a pre-linked version of the kernel that includes all device drivers that are involved in the boot process. This pre-linked kernel is located in /System/Library/Caches/com.apple.kernelcaches. By linking these drivers into the kernel ahead of time, boot time is reduced.

If the pre-linked kernel is missing, out-of-date, or corrupt, the boot loader attempts to load that same set of device drivers all at once in the form of a single, compressed archive called an mkext cache.

If this cache is also out-of-date, missing, or corrupt, the boot loader searches /System/Library/Extensions for drivers and other kernel extensions whose OSBundleRequired property is set to a value appropriate to the type of boot (for example, local or network boot).

For more information on how drivers are loaded, see *I/O Kit Fundamentals*.

Once the kernel and all drivers necessary for booting are loaded, the boot loader starts the kernel's initialization procedure. At this point, enough drivers are loaded for the kernel to find the root device. Also from this point, on PowerPC-based Macintosh computers, Open Firmware is no longer accessible (quiesced).

The kernel initializes the Mach and BSD data structures and then initializes the I/O Kit. The I/O Kit links the loaded drivers into the kernel, using the device tree to determine which drivers to link. Once the kernel finds the root device, it roots(*) BSD off of it.

> **Note:** As a terminology aside, the term "boot" was historically reserved for loading a bootstrap loader and kernel off of a disk or partition. In more recent years, the usage has evolved to allow a second meaning: the entire process from initial bootstrap until the OS is generally usable by an end user. In this case, the term is used according to the former meaning.
>
> As used here, the term "root" refers to mounting a partition as the root, or top-level, filesystem. Thus, while the OS boots off of the root partition, the kernel roots the OS off of the partition before executing startup scripts from it.

Prior to Mac OS X v10.4, the remaining system initialization was handled by the mach_init and init processes. During the course of initialization, these processes would call various system scripts (including /etc/rc), run startup items, and generally prepare the system for the user. While many of the same scripts and daemons are still run, the mach_init and init processes have been replaced by launchd in Mac OS X v10.4 and later. This change means that launchd is now the root system process.

In addition to initializing the system, the launchd process coordinates the launching of system daemons in an orderly manner. Like the inetd process, launchd launches daemons on-demand. Daemons launched in this manner can shut down during periods of inactivity and be relaunched as needed. (When a subsequent service request comes in, launchd automatically relaunches the daemon to process the request.)

This technique frees up memory and other resources associated with the daemon, which is worthwhile if the daemon is likely to be idle for extended periods of time. More importantly, however, this guarantees that runtime dependencies between daemons are satisfied without the need for manual lists of dependencies.

Next, launchd(8) starts SystemStarter(8), which starts any non-launch-on-demand daemons.

> **Note:** While launchd does support non-launch-on-demand daemons, this use is not recommended. The launchd daemon was designed to remove the need for dependency ordering among daemons. If you do not make your daemon be launch-on-demand, you will have to handle these dependencies in another way, such as by using the legacy startup item mechanism.

For more information about launch-on-demand and SystemStarter daemons and how to launch them, see "Daemons".

As the final part of system initialization, launchd launches loginwindow. The loginwindow program controls several aspects of user sessions and coordinates the display of the login window and the authentication of users.

> **Note:** By default, Mac OS X boots with a graphical boot screen. For debugging the boot process, it is often useful to disable this, revealing the text console underneath. This mode is known as verbose boot mode. To enable verbose boot mode, simply hold down command-v after the boot chime.

Boot EFI Utilities

rEFIt

Apple does not offer any direct tools for accessing EFI. There is no key sequence available to enter EFI upon boot. There are, however, utilities available to access this. One such tool, rEFIt is available on Sourceforge.net. The link at the time of this writing is http://refit.sourceforge.net/

The utility can be run on a Live Macintosh, but is not available without installation. In our case, the more useful option is to boot from a bootable disk with the utility installed and gather the needed information. Typically, this information is the system date and time along with any other low-level information your agency elects to include. You will need to have created a forensically sound boot disk (external hard drive, USB drive, DVD, etc.) and have this tool included.

Because of the lack of EFI documentation, single-user mode is probably the better way to gather information such as date and time at this point.

Booting a Macintosh from the LiveCD

Booting from a LiveCD on a Macintosh is a rather straight-forward process, yet have many different paths that can be followed. We will not be discussing the specific directions for each LiveCD offered on a Macintosh. Your agency should develop specific operating guides for the tool(s) used. An internet search for Knoppix, Linux, and the likes on a Macintosh will yield many variations that might boot the target Macintosh. Be careful when selecting a LiveCD. You want to know what *happens* when the LiveCD is running. Some LiveCDs have the potential to alter the target disk, just as if you booted from the target disk itself. Do not make your first test during an actual limited scope examination.

Some available distributions:

- PowerPC - Ubuntu LiveCD (discontinued development as of 02/2007)
- Intel - Ubuntu LiveCD
- Intel - Helix LiveCD
- PowerPC and Intel - BBT Macquisition CD

From a LiveCD that is Linux based, the DD utility will allow for a bit for bit, forensic copy of the original device. You will need to familiarize yourself with the console and GUI of each distribution. Each will have their own nuances that can potentially change what you are accustomed to seeing as output.

Imaging a Target Macintosh

Once it has been determined that you wish to make an image of the target Macintosh vs. collecting certain files and folders, steps need to be taken to insure the result is as expected. The steps that need to be taken will highly depend on the method/path chosen. We will deal with this here. We are going to use in this outline, the tools available from the typical install, and NOT specialized, downloaded tools. There are tools that will make some of these steps easier, or in fact combine the steps creating shorter acquisition times altogether. Explore these tools after you are comfortable with the well-known, established results of the steps taken here.

Target Disk Mode

In target disk mode, the target computer acts as an external firewire hard drive. The steps to acquire such a device are the same as any other firewire hard drive. Windows will alter a Macintosh in this mode if any writable partitions exist (FAT32, NTFS). Because of this, and the lack of up-front knowledge of whether or not these exist, it is recommended an acquisition of this type be done with a forensic Macintosh. It is also possible to use Linux and image the drive with DD (disk dump). The procedure varies only slightly.

The specific steps for a Target Disk Mode acquisition with a forensic Macintosh are as follows:

1. Turn off DiskArbitration on your forensic Macintosh (alternately, use a specific partition on your forensic Macintosh that always has DiskArbitration off) [see Activate/Deactivate Disk Arbitration]
2. Shut down your forensic Macintosh.
3. Start the target Macintosh following the Target Disk Mode Procedure outlined earlier.
4. Connect the target Macintosh to your forensic Macintosh via a firewire cable.
5. Boot your forensic Macintosh either to your forensic partition or with DiskArbitration turned off.
6. If all is well, you will see your boot drive on the desktop, but nothing else (because DiskArbitration is off).
7. Enter the Terminal and check for your attached Target Disk Mode Macintosh

   "`hdiutil info`" will yield device information [or]

   "`ls /dev/disk?`" to get a listing of recognized devices
8. Determine which disk you will acquire and create a digital fingerprint of the target device by running MD5 hash. Assuming the disk you will acquire is disk1, use the MD5 command as follows:

   "`md5 /dev/disk0 > /Evidence/targetMacintosh.md5_start`"
9. A "raw" disk, or rdisk, will acquire faster than is buffered disk counterpart. Assuming the disk you will acquire is "disk1", use dd to make the acquisition of the raw disk as follows:

"`dd if=/dev/rdisk1 conv=noerror,sync of=/Evidence/targetMacintosh.dd`"

10. The dd utility will not give an progress reporting, and will simply exit when it is finished. A notice on screen stating the number of blocks in and blocks out will be reported. They should match if everything was copied bit for bit as expected.
11. Create a second digital fingerprint of the target device to show nothing has been altered by the dd process.

    "`md5 /dev/disk0 > /Evidence/targetMacintosh.md5_end`"
12. Power down your forensic Macintosh.
13. Power down the target Macintosh by holding down it's Power button.
14. Disconnect the firewire cable and you are finished.

Possible failures of this method include: lack of drive space on your forensic Macintosh to acquire, faulty firewire cable, or a physically failing target Macintosh.

Other tools to consider for this method would include DCFLDD and BBT Forensic Suite.

LiveCD

A LiveCD method for acquisition of a Macintosh is sometimes the preferred method. This involves booting the target Macintosh with a known, forensically sound CD. LiveCD's can include a custom tailored Linux distribution such as Helix, SMART or a Knoppix variant. It can also include paid-for tools like BBT Macquisition.

Drive Removal

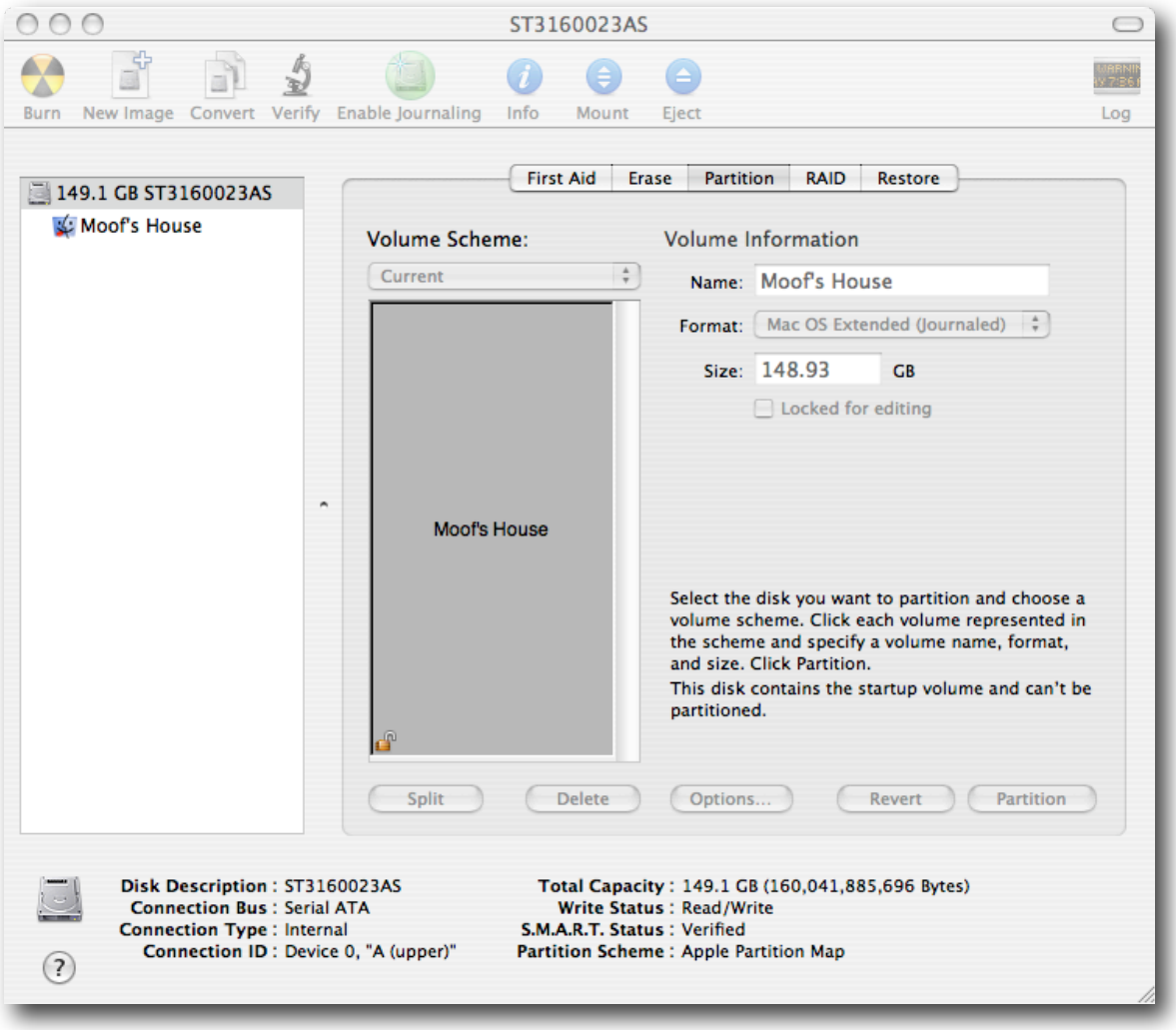
Physical drive removal sometimes is the most complicated part of a Macintosh examination. The cases of some Macintosh computers will seem like a security barrier as you try to open them. Others will open within seconds and present the internal drives very neatly. When choosing this method, you will likely want to use a physical write blocking device for the acquisition. Many companies offer a great selection of just such devices. The appropriate steps to take will be determined by the physical write blocking device you choose to use. Once the disk drive is physically write blocked, an imaging process can begin with any tool of your choosing, on any operating system.

Possible failures of this method include: a bad cable between the drive and the physical write blocking device, bad cable from the physical write blocking device to the forensic computer, and the imaging tool can't recognize the file system of the target Macintosh hard drive and displays the disk as unallocated space.

Disk Structure

Apple Partition Map

Macintosh computers will likely use one of two partitioning schemes. From the factory, PowerPC based Macintoshes come with the Apple Partition Map. An Intel based Macintosh, however, will utilize the new GUID partition scheme. Do not confuse this with the file system of HFS or HFS+. The partitioning scheme is the basic definition of how a hard drive or other media is laid out for a file system to be applied. Here is a look at the disk structure of a typical PowerPC based Macintosh:



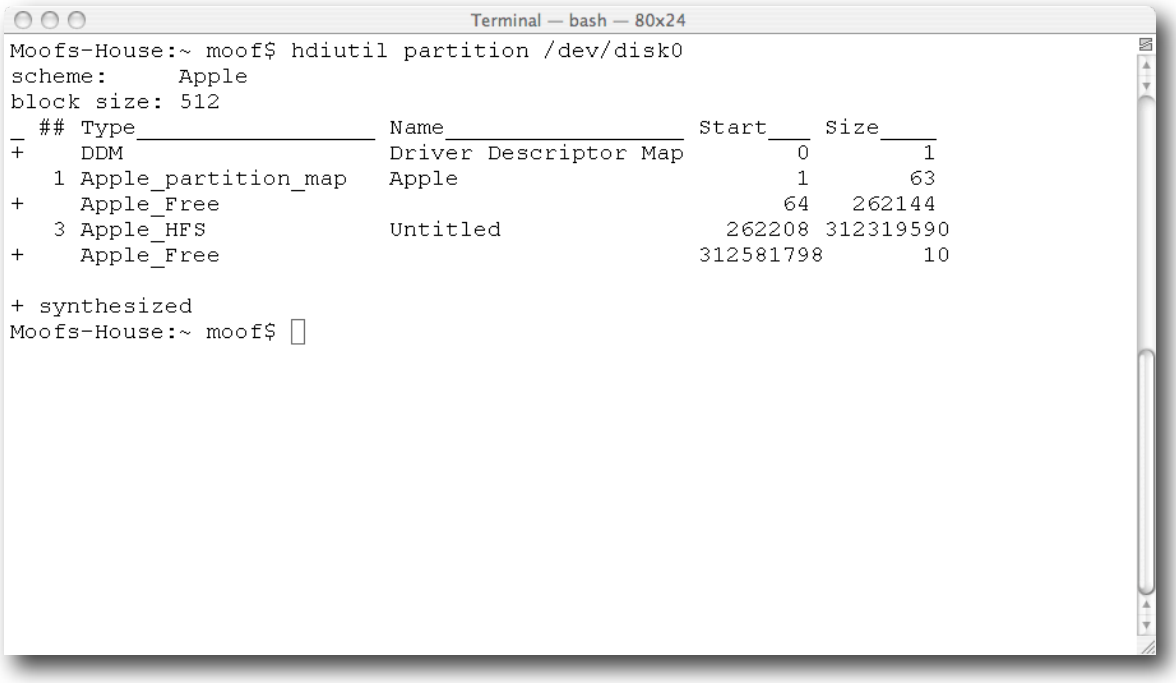
Disk Utility - Apple Partition Map

The image shows a 149.1 GB hard drive with model number ST3160023AS with a user given name of "Moof's House". The Volume Scheme shows the drive having only one partition, and the format

used is Mac OS Extended (Journaled). Note at the bottom, Apple Partition Map is the partition scheme used. What does all of this mean?

The left window pane shows us physical storage devices. Physical storage could also include a DMG that has been mounted as well. On this computer, only one hard drive is connected. Looking at the lower portion of the window, the drive is a Serial ATA or SATA drive. The Volume Scheme section gives information on the number and types of partitions available. The current partition map shows one large partition across the entire available drive. It has been named "Moof's House" and is formatted using HFS+ with journaling enabled. More to come on Journaling later.

Now, let's look at the same disk through the Terminal window using "hdiutil".


Terminal — bash — 80x24
Moofs-House:~ moof$ hdiutil partition /dev/disk0
scheme: Apple
block size: 512
_ ## Type Name Start Size
+ DDM Driver Descriptor Map 0 1
1 Apple_partition_map Apple 1 63
+ Apple_Free 64 262144
3 Apple_HFS Untitled 262208 312319590
+ Apple_Free 312581798 10
+ synthesized
Moofs-House:~ moof$

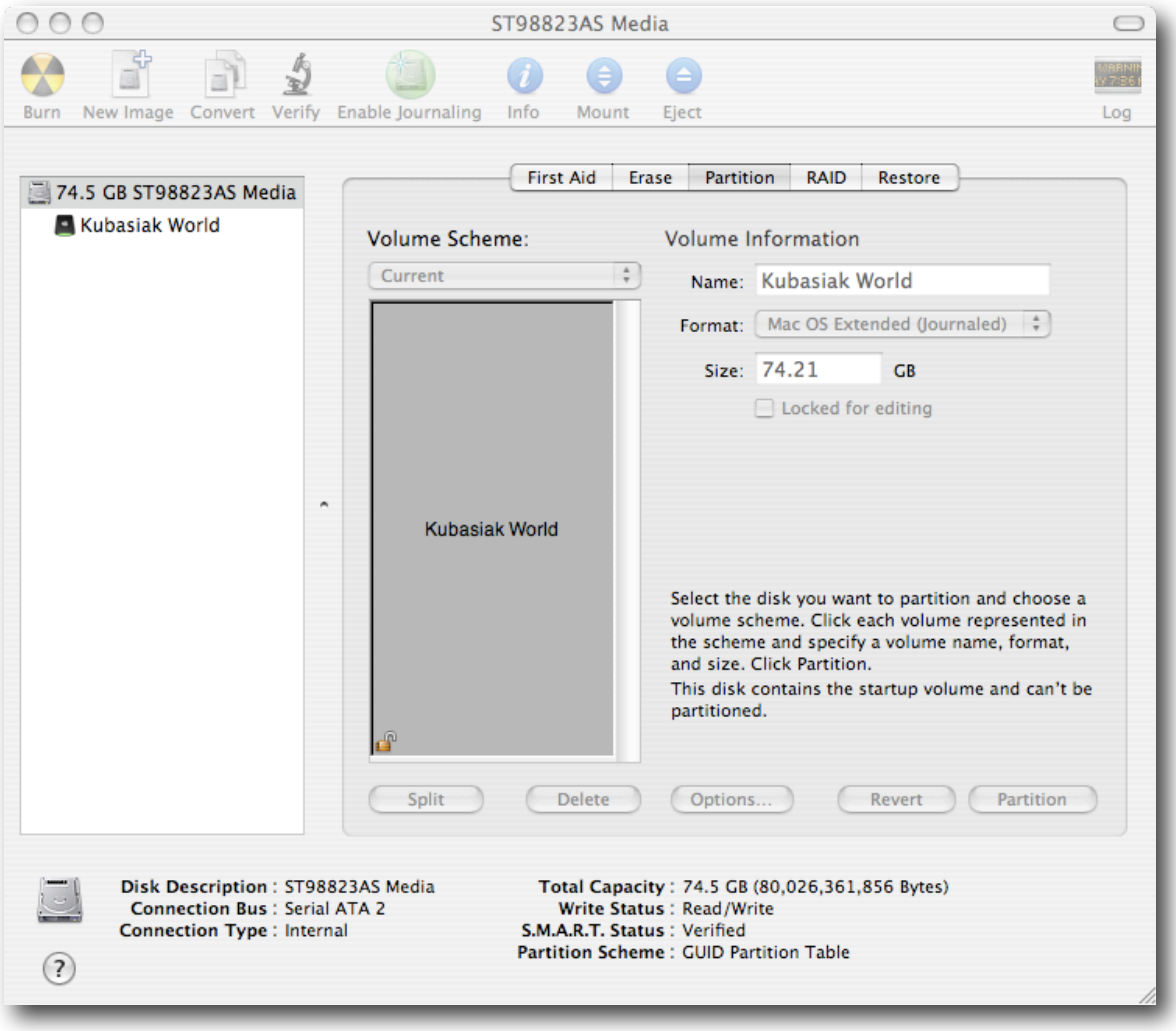
Terminal Window - Apple Partition Map

The command used to give this view was "hdiutil partition /dev/disk0". Notice the extra information we are now seeing as compared to the output of Disk Utility. Sector 0 is the boot sector with a size of 1 sector. Sectors 1 thru 64 is the Apple Partition Map defining the layout of the disk. Apple Free is a "padding" defined as being available for future use. The data section for a forensic analysis finally shows up at the Apple HFS partition starting at sector 262208 and having a length of 3,122,319,590 sectors. There is one more Apple Free partition with a length of 10 sectors, again used as padding.

The Apple Free area is not normally where data will be found. It is not easily accessed by the casual user. However, nothing prevents a more savvy user from hiding information there with the right tools. Also, information could be left over in these areas from a previous partition scheme.

GUID Partition Table

Next, let's look at an Intel based Macintosh. Here is the Disk Utility information window.



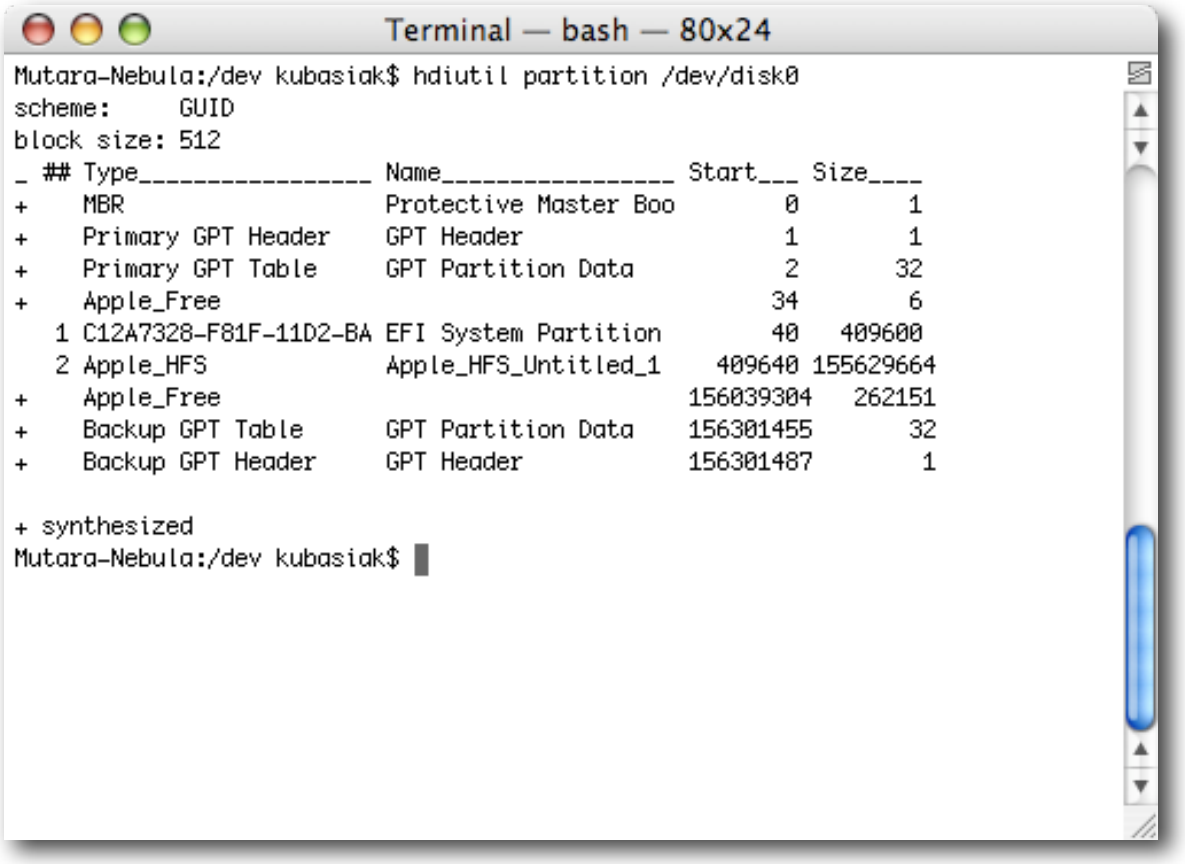
Disk Utility - GUID Partition Table

The image shows a 74.5 GB hard drive with model number ST98823AS with a user given name of "Kubasiak World". The Volume Scheme shows the drive having only one partition, and the format used is Mac OS Extended (Journaled). Note at the bottom, GUID Partition Table is the partition scheme used. What does all of this mean?

The left window pane shows us physical storage devices. On this computer, only one hard drive is connected. Looking at the lower portion of the window, the drive is a Serial ATA 2 or SATA2 drive.

The Volume Scheme section gives information on the number and types of partitions available. The current partition map shows one large partition across the entire available drive. It has been named "Kubasiak World" and has been formatted using HFS+ with journaling enabled. Again, more to come on Journaling later.

Now, let's look at the same disk through the Terminal window using "hdiutil".


Terminal — bash — 80x24
Mutara-Nebula:/dev kubasiak$ hdiutil partition /dev/disk0
scheme: GUID
block size: 512
_ ## Type_____ Name_____ Start___ Size____
+ MBR Protective Master Boo 0 1
+ Primary GPT Header GPT Header 1 1
+ Primary GPT Table GPT Partition Data 2 32
+ Apple_Free 34 6
1 C12A7328-F81F-11D2-BA EFI System Partition 40 409600
2 Apple_HFS Apple_HFS_Untitled_1 409640 155629664
+ Apple_Free 156039304 262151
+ Backup GPT Table GPT Partition Data 156301455 32
+ Backup GPT Header GPT Header 156301487 1
+ synthesized
Mutara-Nebula:/dev kubasiak$

Terminal Window - GUID Partition Table

The command used to give this view was "hdiutil partition /dev/disk0". Notice the extra information we are now seeing as compared to the output of Disk Utility. Sector 0 is the boot sector with a size of 1 sector. Sector 1 is the Primary GUID Partition Table Header and sector 2 thru 34 contains GUID Partition Table data defining the layout. Notice that these two partitions are replicated at the end of the drive in reverse order. We will recognize the Apple Free partition and the function is similar in nature. The data we are interested in for an exam lies within the Apple HFS partition starting at sector 409,640.

For an even more in-depth look at this topic, read "Technical Note 2166 - Secrets of the GPT" on the Apple Developer website.

Journaling

(Apple Document 107249)

"Journaling" is a feature that helps protect the file system against power outages or hardware component failures, reducing the need for repairs. Journaling was first introduced in Mac OS X Server 10.2.2, then to the non-server OS in Mac OS X 10.3 Panther. This document explains some of the benefits of using this feature and how it works.

Journaling for the Mac OS Extended (HFS Plus) file system enhances computer availability and fault resilience, which is especially noteworthy for servers. Journaling protects the integrity of the file system on Xserve and other computers using Mac OS X Server in the event of an unplanned shutdown or power failure. It also helps to maximize the uptime of servers and connected storage devices by expediting repairs to the affected volumes when the system restarts.

Journaling is a technique that helps protect the integrity of the Mac OS Extended file systems on Mac OS X volumes. It both prevents a disk from getting into an inconsistent state and expedites disk repair if the server fails.

When you enable journaling on a disk, a continuous record of changes to files on the disk is maintained in the journal. If your computer stops because of a power failure or some other issue, the journal is used to restore the disk to a known-good state when the server restarts.

With journaling turned on, the file system logs transactions as they occur. If the server fails in the middle of an operation, the file system can "replay" the information in its log and complete the operation when the server restarts.

Although you may experience loss of user data that was buffered at the time of the failure, the file system is returned to a consistent state. In addition, restarting the computer is much faster. Always remember to back up your data as frequently as necessary.

What does this mean for us as digital forensic investigators? Two thoughts need to be considered with every case:

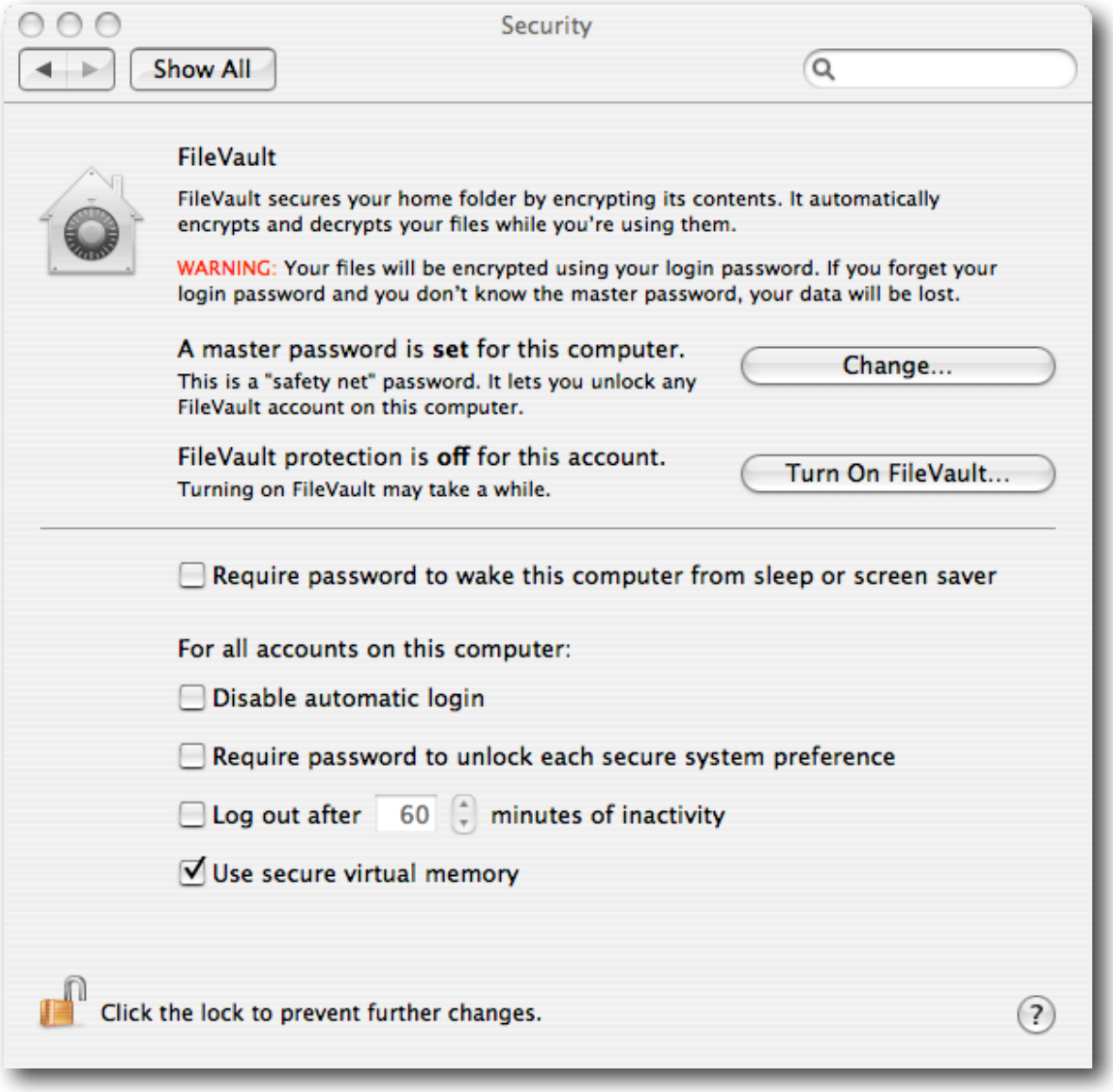
- Do I shut down this Macintosh normally or pull the plug?
- Booting a forensically restored version of a Macintosh that has journaling will result in automatic correction to corruption.

The answers to these these questions will depend on how you or your agency establish policies.

FileVault and MacOS X Security

FileVault Preference Pane

FileVault is the security technology available in MacOS 10.4 to secure a user's home directory. When turned on, the user's home directory will be encrypted using 128 bit AES encryption to a Sparseimage DMG file.


Security
Show All
FileVault
FileVault secures your home folder by encrypting its contents. It automatically encrypts and decrypts your files while you're using them.
WARNING: Your files will be encrypted using your login password. If you forget your login password and you don't know the master password, your data will be lost.
A master password is set for this computer.
This is a "safety net" password. It lets you unlock any FileVault account on this computer.
Change...
FileVault protection is off for this account.
Turning on FileVault may take a while.
Turn On FileVault...
Require password to wake this computer from sleep or screen saver
For all accounts on this computer:
Disable automatic login
Require password to unlock each secure system preference
Log out after 60 minutes of inactivity
Use secure virtual memory
Click the lock to prevent further changes.
?

Security Preference Pane

The window shows the available security features from the Security Preference Pane. A description of each follows.

Master Password - This is the master password used to unlock a FileVault sparseimage when the user has forgotten the password.

Turn On FileVault - Clicking on this button will enable FileVault for the currently logged in account. The sparseimage of the user's home directory will be created and the user will be logged out.

Require password to wake this computer from sleep or screen saver - will cause the computer to prompt for the currently logged in user's password to wake or unlock the screen saver

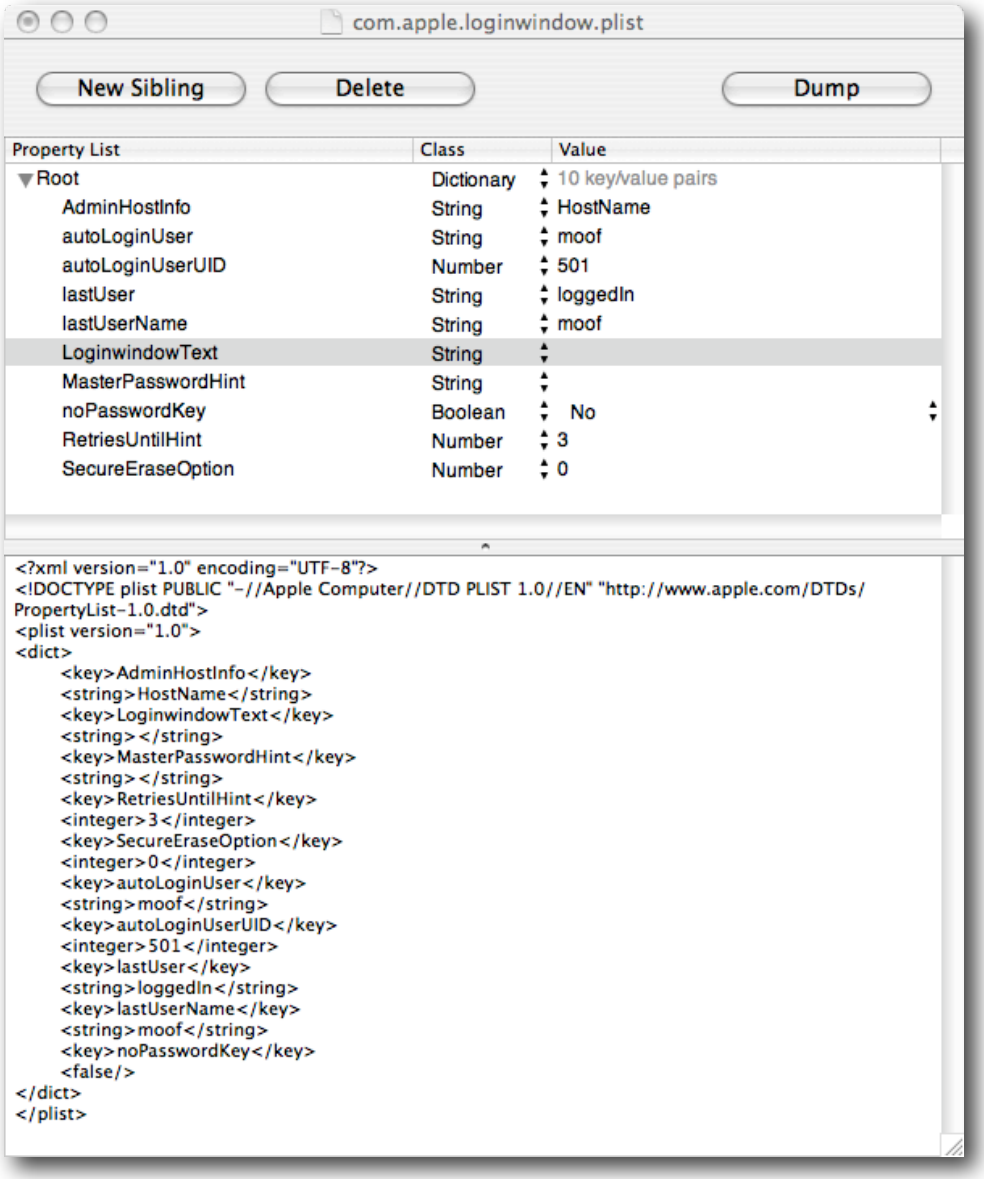
Disable automatic login - Causes the Login Window to appear during the boot sequence. When this is not checked, the selected user will automatically login during the boot sequence.

Require password to unlock each secure system preference - Forces a password to be entered before changes to security can be made

Log out after X minutes of inactivity - Will cause automatic log of the currently logged in user (or users) after the specified number of minutes

Use secure virtual memory - causes the /var/vm/swapfile0 and other subsequent page files to be encrypted. When this is not checked, all pages of memory to disk are in clear text, offering an abundant source of user information. The swapfiles are deleted during boot, and NOT at shutdown or logout!

It is important for a full analysis to include items such as the options listed above. For instance, it is not the same story when a system has the auto-login feature on vs. off. Having to know a password to get into the system narrows down the number of people that may have used a computer immediately. In order to gain this information, "plist" files will need to be examined. A likely area for system-wide setting to be stored is /Library/Preferences. Here is an example of the *loginwindow* plist file.

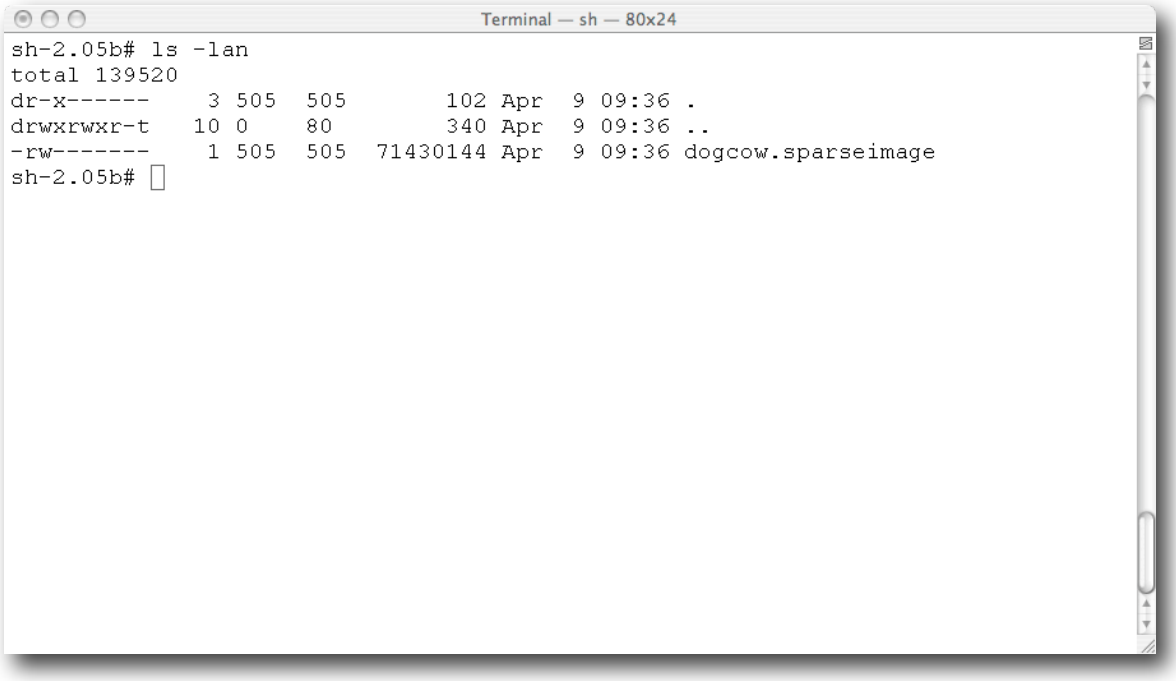


Property List Editor - com.apple.loginwindow.plist

Here we see that the Auto Login setting has been set, and the user *moof* will be used (UID 501).

sparseimage and User Home Directory

FileVault and the sparseimage file created is simply a DMG file that has been encrypted with 128 bit AES encryption using the user's login password. A sparseimage also will expand and compact as size requirements change for the "disk". That is different from a DMG where the entire size is allocated up front. It will be named *username*.sparseimage and will be located in the user's home directory. This file can be manipulated like any other file, and can be successfully mounted if the password is known. As with any DMG file, you should "Lock" the file before using it. This will ensure Read-Only privileges regardless of the level of account being used. Even "root" will not have privilege to write to this file when the HFS+ "Lock" is used. Here is a screen capture of a user's home directory after FileVault has been turned on.


Terminal — sh — 80x24
sh-2.05b# ls -lan
total 139520
dr-x------   3 505 505   102 Apr 9 09:36 .
drwxrwxr-t   10 0   80   340 Apr 9 09:36 ..
-rw-------   1 505 505 71430144 Apr 9 09:36 dogcow.sparseimage
sh-2.05b#

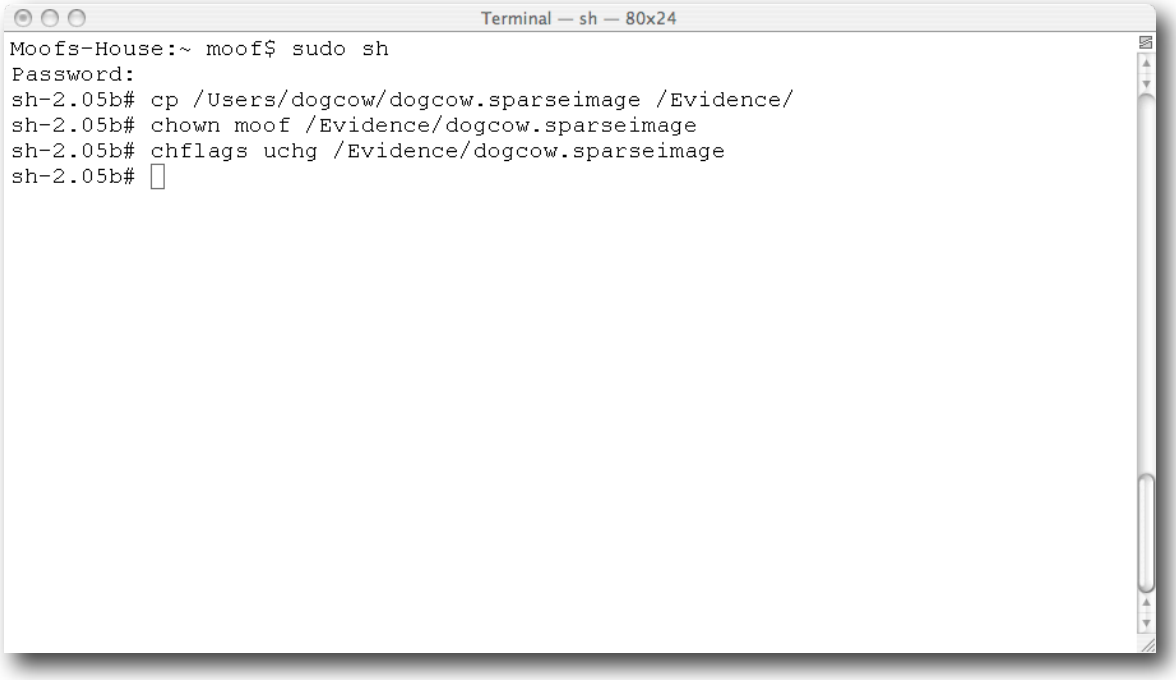
Terminal Window - User Home Directory with FileVault Enabled

In this example, the user *dogcow* has FileVault turned on. The home directory now contains only a single file, *dogcow*.sparseimage is a DMG sparseimage that has been 128 bit AES encrypted. You can see that user 505 is the owner and its size is currently 71,430,144 bytes large. This file can be copied to another drive as any other file could. You will need an admin account to access another user's directory, and you will need the encryption password (login password) of the user *dogcow* to mount this file.

Acquire the Encrypted User Home Directory

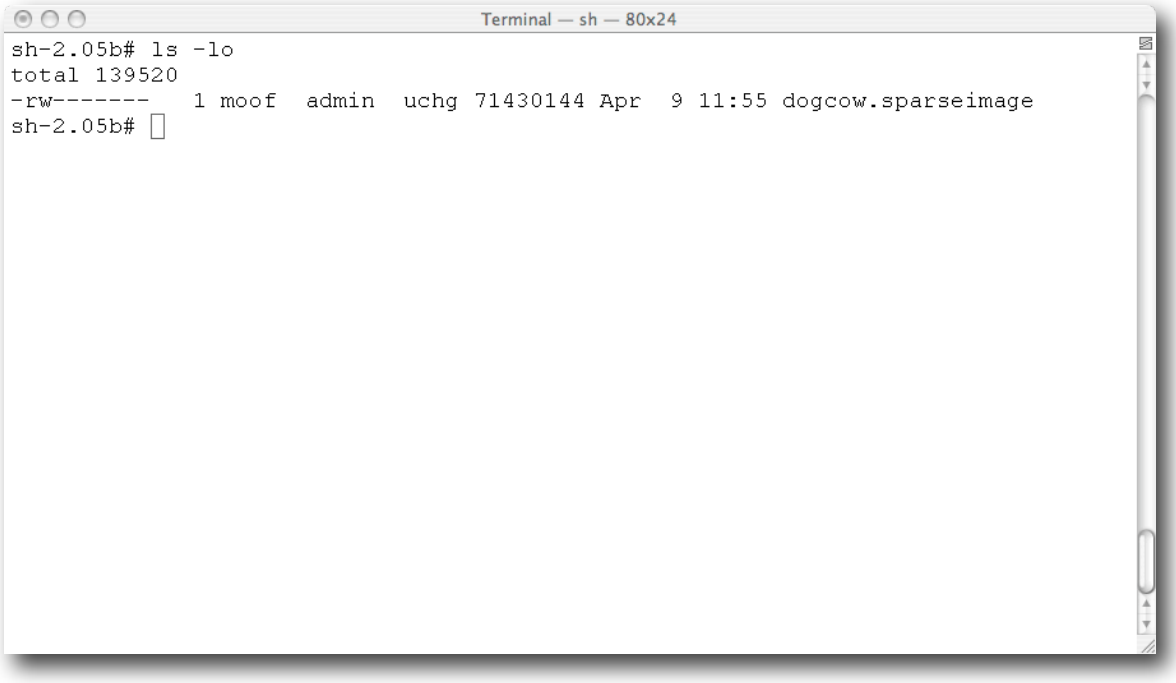
When copying this file, do not forget to immediately set the "Locked" property in the Finder. This will prevent any changes occurring to the file. Here are the steps to successfully acquire this file.

1. Open a shell in the terminal with root privileges. (BE CAREFUL!)
    - Example "sudo sh"
2. Copy the file from its present location to your Evidence Collection directory.
    - Example "cp /Users/dogcow/dogcow.sparseimage /Evidence"
3. Take ownership of the file.
    - Example "chown *yourusername* /Evidence/dogcow.sparseimage"
4. Set the Locked flag to prevent any changes to this file.
    - Example "chflags uchg /Evidence/dogcow.sparseimage"



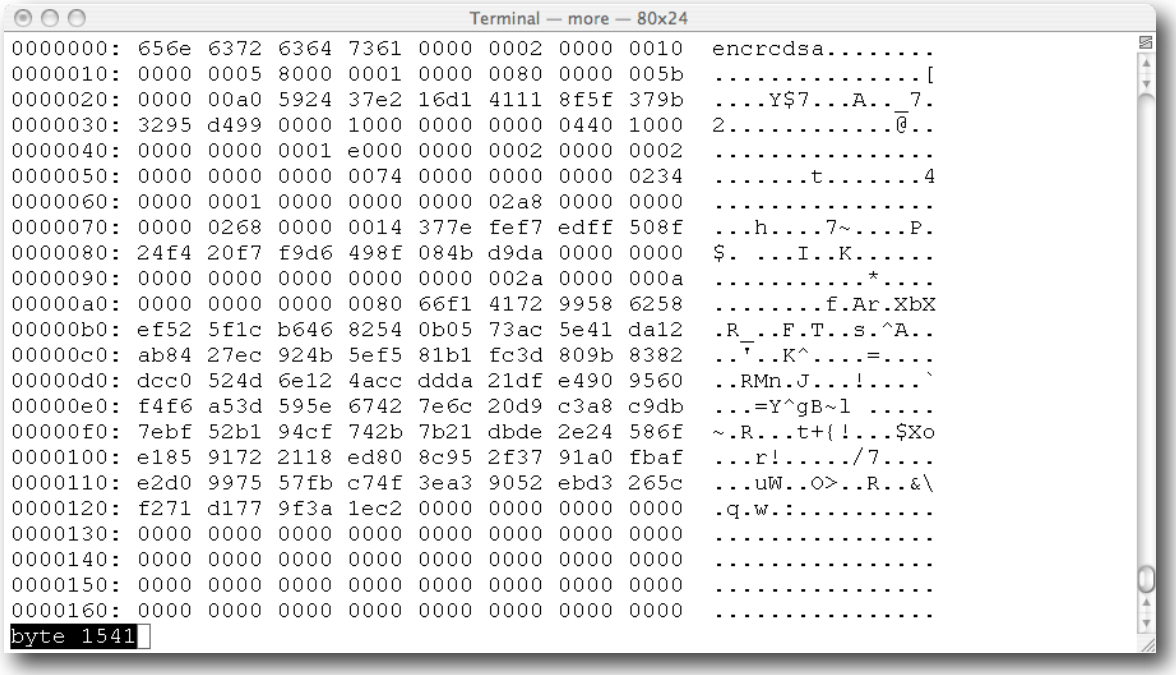
Terminal Window - Forensically Copy sparseimage

Looking at the /Evidence directory after the steps have been completed will result in the following output.



Terminal Window - Attributes of sparseimage Properly Copied

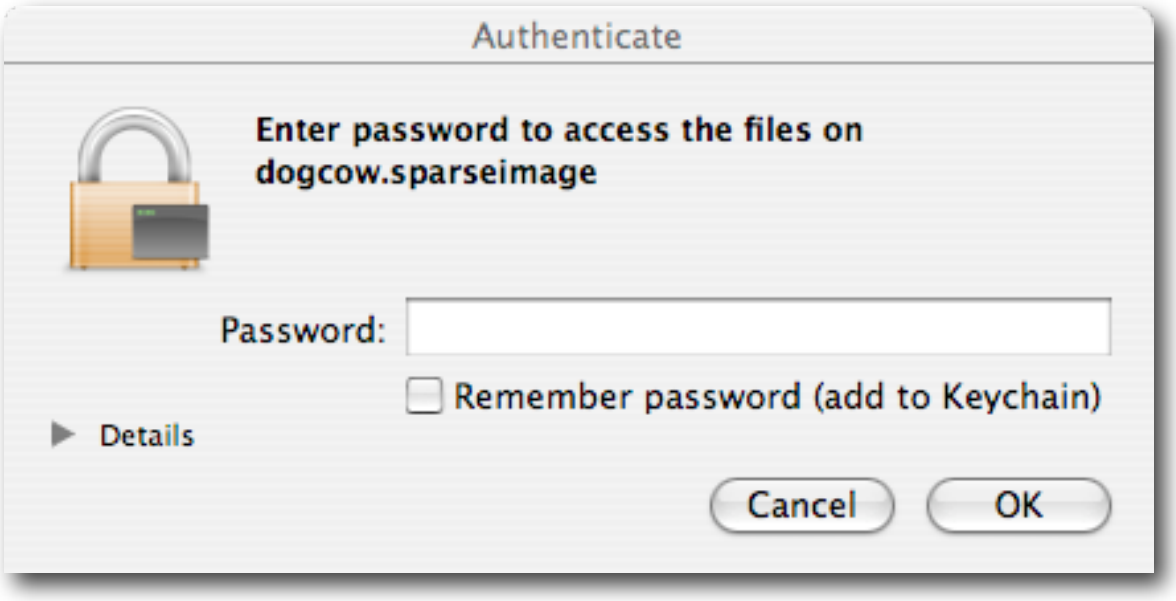
Prior to mounting the sparseimage, looking at the contents will result in nothing but gibberish. Nothing useful can be gathered from the image itself except for one fact. The header of a sparseimage will will show as follows:



Terminal - XXD View of sparseimage Header

Every sparseimage will have the header “encrcdsa”.

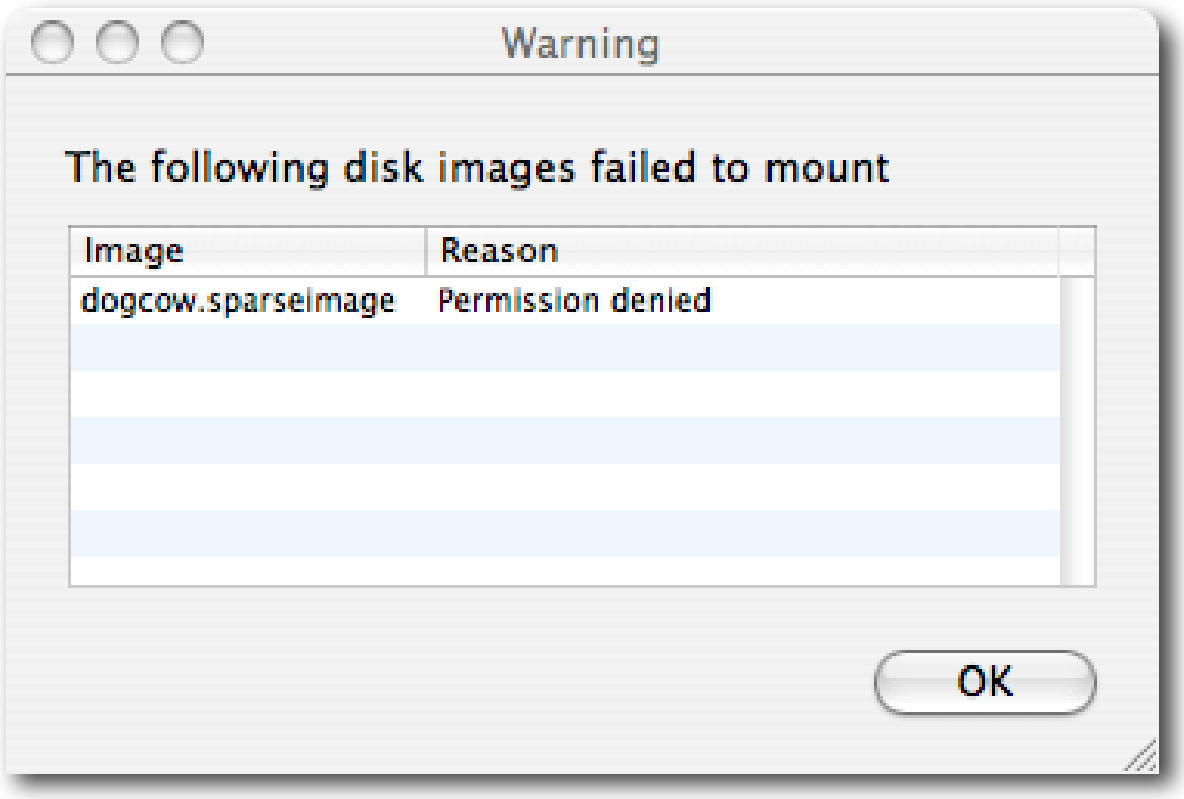
You should now be able to mount the Disk Image file in your evidence directory. If you have taken each of the steps from above, double-clicking on the file will result in the following dialog:



Finder - Authentication Dialog

Entering the login password for *dogcow* will result in the image mounting on your desktop.

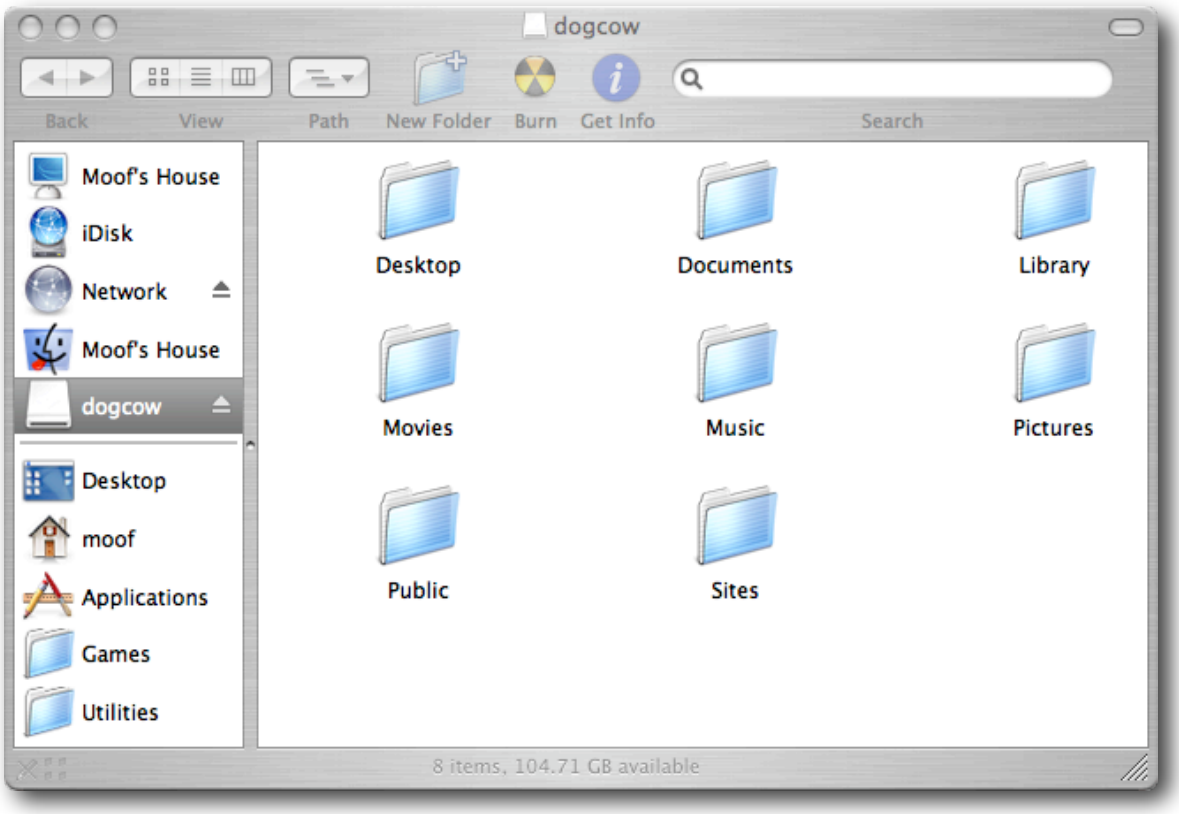
If you get the following dialog instead:



Finder - Image Mounting Error Dialog

then you have not appropriately taken ownership of the image file with the “chown” command.

Once the Disk Image is mounted, you will recognize the user's home directory:



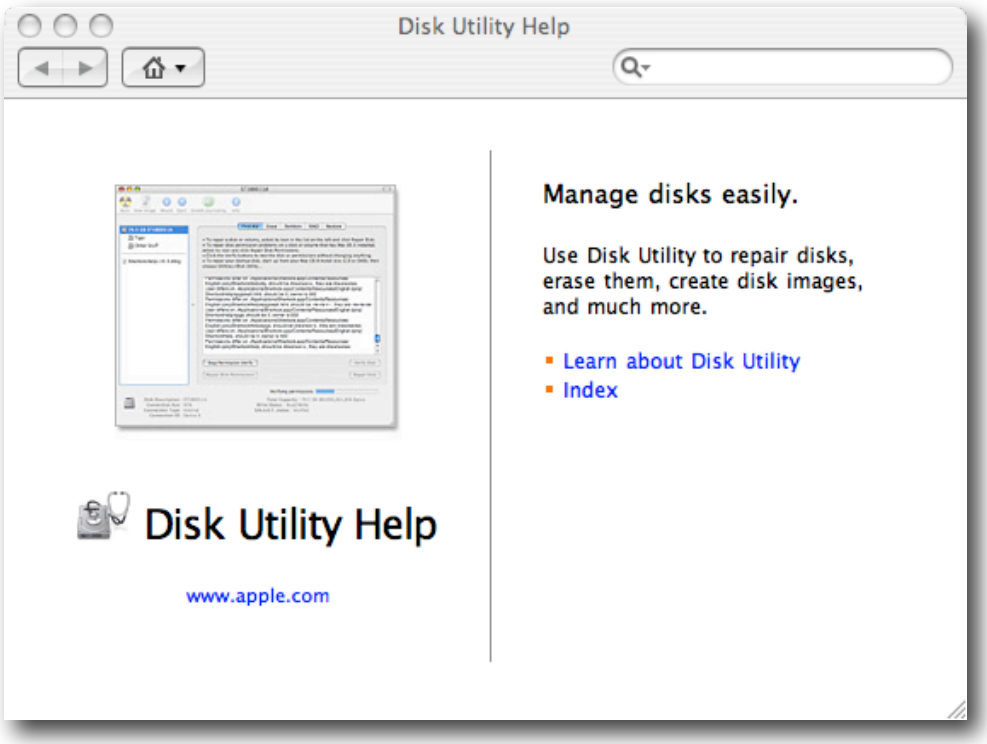
Finder - *dogcow* User Home Directory

The image is locked, and can be verified with a Get Info from the Finder. All searches and file examinations are read-only at this point.

To be more complete in this examination, hash values should be computed prior to mounting the image file, and after ejecting the image file.

DiskUtility and DMG Files

DiskUtility Features


Disk Utility Help
Manage disks easily.
Use Disk Utility to repair disks, erase them, create disk images, and much more.
Learn about Disk Utility
Index
Disk Utility Help
www.apple.com

Disk Utility - Help Window

Disk Utility is a powerful application included with every Macintosh running MacOS X. If you haven't already, look at the Help file for this program and familiarize yourself with its many function. We are going to talk about a few specific areas of forensic value in this application. In order for Disk Utility to function, DiskArbitration needs to be enabled. As an examiner, you will want to have acquired your image of the target first, then, with your examination computer, you can re-enable DiskArbitration.

DMG vs. sparseimage

There are many types of "image" files. DD is a UNIX (or linux) utility that creates a Disk Dump of the given device. Guidance Software's Encase creates E01 files for it's output. Disk Utility natively will create the DMG file or Disk Image file, as well as the sparseimage file. It has the ability to deal with many other file types that will not be dealt with here. The MAN page on hdiutil will give you a wealth of knowledge on current and historical image file types. The biggest difference between the DMG file and sparseimage file is initial file size. A DMG file will have the file size allocated up front in creation. For instance, when creating a DMG file of size 40MB, 40MB of disk space will used right away. When creating the sparseimage file of 40MB in size, about

10MB will be used initially, and the file will grow (or shrink) as necessary up to the maximum of 40MB.

Encrypted vs. Unencrypted

The encryption provided thru the Disk Utility program is AES 128 bit encryption. It is used by default when a user's home directory is encrypted with FileVault, and can also be selected during the creation of a DMG file. An encrypted DMG file or sparseimage file is near useless in today's computing environment. A brute force attack with a dictionary or rainbow table *may* yield good results, but likely will give you what you started with, nothing. This section is not written to discourage you from obtaining encrypted DMG or sparseimage files. It is to encourage you to pursue other investigative measures in obtaining the password. The best encryption in the world is easily cracked with the password written on a sticky note.

DD and Raw Images

DD or Disk Dump is an old UNIX utility that was used to back up systems to tape drives originally. It turns out that DD creates a forensically sound image of a device for us. There are specialized versions of this program, such as DCFLDD that extend the original capabilities of DD. Many programs use DD as their underlying basis of operation. A DD image file is considered a raw image because it will match the original device, but for bit, with no compression. Mounting a DD image file for analysis will show that the file is indistinguishable from the original and will produce the same MD5 hash value. There is a known flaw with DD running on linux with certain versions of the kernel code. The flaw simply causes DD to miss the last sector of some odd-total-sector drives. This is rare to find, but worth noting in this section.

Spotlight

(Apple Document 301533)

The information here comes from the best source, Apple Inc. The following information is directly from the Support website.

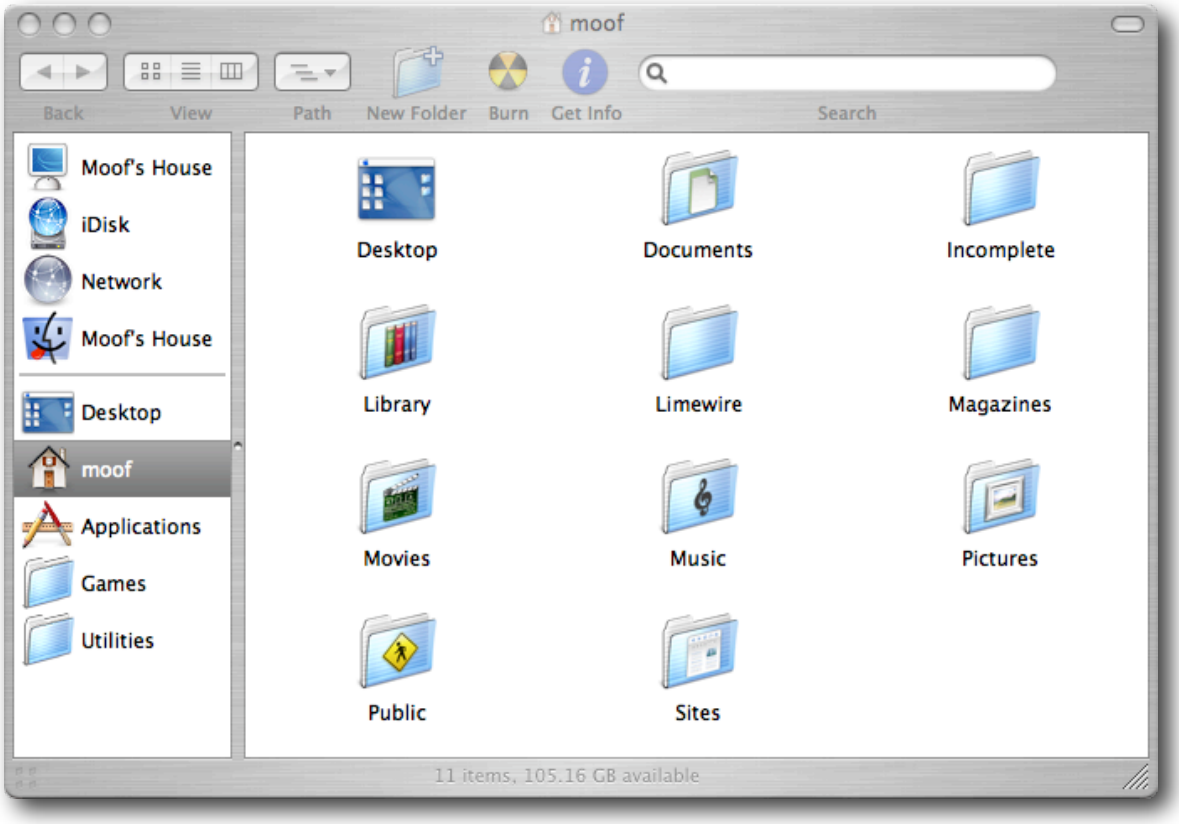
Mac OS X 10.4 Tiger features Spotlight, a lightning-fast search technology that instantly lets you find things on your Mac. By default, Spotlight will index and search in the following locations:

- All Home folders (local and network-based, as well as FileVault and non-FileVault). This includes:
- The Documents, Movies, Music, and Pictures folders
- The Trash of all users and each mounted volume.
- ~/Library/Metadata/
- ~/Library/Caches/Metadata/
- ~/Library/Mail/
- ~/Library/Caches/com.apple.AddressBook/Metadata/
- ~/Library/PreferencePanes/
-
- Spotlight also searches these non-Home folder locations by default:
- /Library/PreferencePanes/
- /System/Library/PreferencePanes/
- /Applications

Can Spotlight search anywhere else? Of course! Any new folder you create in your Home automatically gets indexed so that it's searchable. If you connect an external storage device, such as a USB or FireWire hard drive, Spotlight will index the stuff on it, too. (If you want to exclude certain areas from Spotlight searching, see the tip below.)

Note: If your computer has multiple user accounts, any files that reside at the top level of each user's Home folder will also be indexed and searchable by Spotlight, even though they cannot be modified. However, all files and folders located within a user's Desktop, Documents, Library, Music, Movies, and Pictures folders will not be indexed nor can they be searched by other user accounts using Spotlight.

User Home Directory Structure



Finder - User Home Directory Structure

The home directory is the likely area to find all of the evidence for any case, barring system wide log and settings files. MacOS X is very good at containing a user's files and settings to this area. This trait allows FileVault to work as well as it does. When conducting a limited scope examination, directing your searches to this area first is a good idea.

A User's home directory will contain many standard folder's from a MacOS X installation, as well as application specific folders. The above window shows the user "Moof" home directory. Always remember when using the Finder, the window will NOT show hidden files or directories with the typical MacOS X settings. There is no easy way to change this from any menu, and is best accomplished with a third party application (Onyx, Tinkertool, etc.) or at the command line with a write to the proper Plist file. A description of each entry in the window follows.

Desktop - contains all of the items that are seen on the user's desktop.

Documents - typically will contain user data files such as Pages, Keynote, MS Word, and other types of files.

Incomplete - created by Limewire and will contain files that have not yet successfully downloaded to this user's account. 2 files, downloads.dat and downloads.bak will potentially contain incriminating evidence in the user's use of Limewire

Library - This is a gold mine of information on the way a user utilizes the Macintosh. It will contain logs, preferences, browser history, recent files, etc. Many of these aspects will be discussed in greater detail later.

Limewire - This is created by the Limewire application. By default, shared files and downloaded files will be here. A user can change this location within the application itself.

Magazines - used by the Zinio Reader application for electronic magazines

Movies - typically will contain iDVD movie data, Quicktime files, and other digital video material

Music - typically will contain a user's iTunes library and other digital music material such as MP3 files.

Pictures - typically will contains a user's digital photo collection such as the iPhoto library.

Public - this is a "drop box" where other users have permissions to place files, read files, but not delete files.

Sites - if a WWW server is active such as the built in Apache web server, a user can host their web site from this directory. This may contain a user's internet published incriminating evidence.

User Library Folder - In Depth

The User Library folder will contain huge amount of information including user specific drivers, fonts, settings, system add-ons, etc. Not everything here will be meaningful to a case. On the other-hand, many items in here will be **direct evidence** of the crimes at hand. Browser history, we page cache, email remnants, email attachments, and indexes are just a few examples of this. My personal Library folder contains 45 folders. Some folders are from a standard MacOS X installation, whereas others are created by installing an application. Here are some of the folders and the information that can be gathered from them.

Application Support - Folders will be located in here that are created from Application installations. When a user removes the application from the system, the folder will remain in here. A manual delete is required to remove this information. Although there may not be specific history here, it will be indicative of an application having been installed, and may show usage information.

Automator - User specific actions will be stored here. The actions are added by the user, and may contain some very indicative information of file copying, server connections and other actions a user wants to automate.

Caches - This folder has the potential to be a gold mine of historical data for the examiner. The contents include information of application usage, web sites visited, buddy lists, downloaded files, etc. The best general advice that can be given regarding this directory is - explore. Look in the folders here and see how the information may apply to your specific case. Keep in mind that many folders here will remain even after an application has been removed from the system!

Cookies - Used by Safari and other web browsers for the Cookies of various websites. A file named "Cookies.plist" is likely in this folder.

Favorites - This folder contains favorites for the "Connect to Server" option in the Finder. It will show other network resources that the User considered important enough to be able to easily return to.

Logs - This folder contains log files for many applications and usage information. Excellent evidentiary resource.

Mail and Mail Downloads - These folders contain email and files that were attached to emails received under this account.

Phones - This folder contains cell phones that have been connected to this computer under this account. Specific information about the phones can be found within the Info.plist file.

Recent Servers - This folder contains information on servers that have been recently connected to including AFP and FTP sites.

Safari - This folder contains the vital information on Safari usage including bookmarks, history, etc.

Each of these folders, and others, should be explored for evidence relating to the specific case at hand. It would be impossible to write specific information for each of the folders and files that can possibly be found here.

Applications

Address Book

Address Book is the bundled application that allows users to store names, addresses, telephone numbers, screen names, web page information and just about anything else related to a contact. Address Book is integrated into many applications, such as Mail, Safari, and .Mac. A user can export VCards from here as well.

iCal

iCal is the bundled calendar application. iCal is a simple program compared to many of the more robust, enterprise type calendar systems. iCal is well used, and has the ability to synchronize with .Mac. A user can also publish a calendar to .Mac for public viewing.

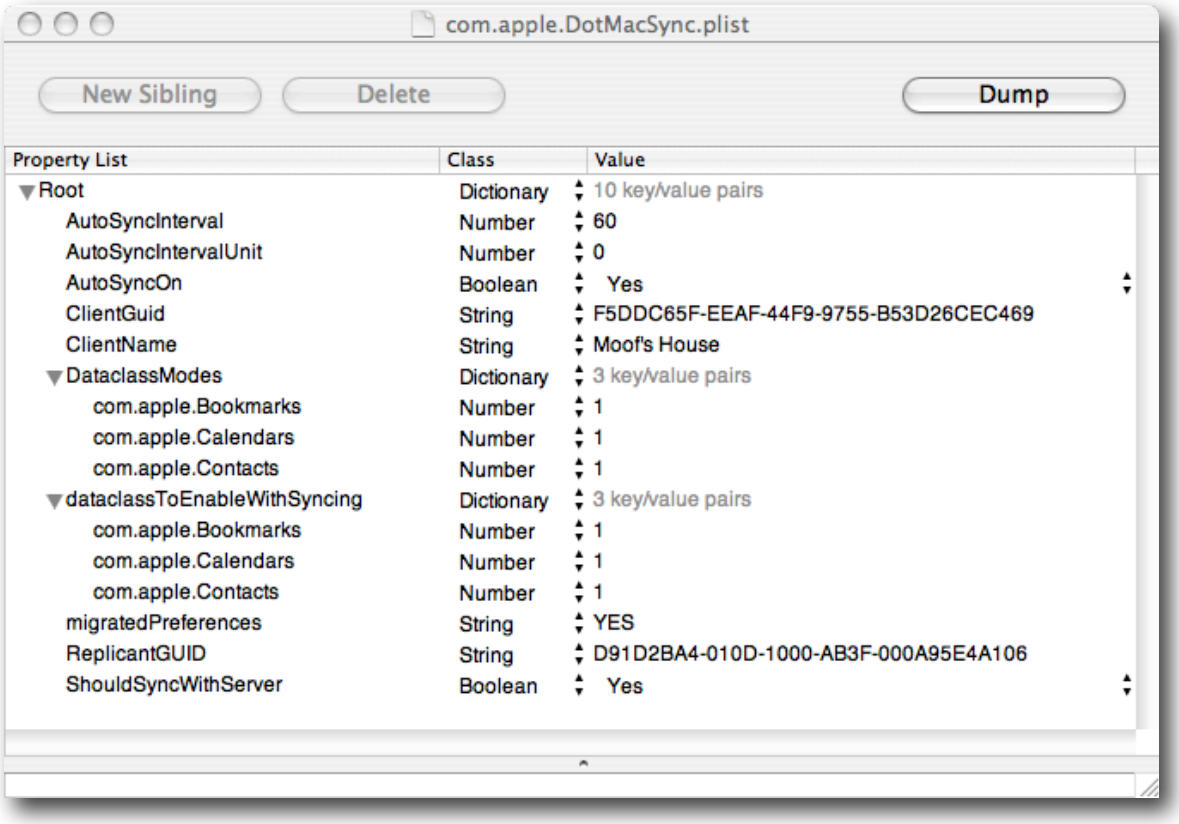
Mail

Mail (or Mail.app as some will call it) is the bundled email application. Mail is integrated with the Address Book, and also maintains a list of people emailed outside of the Address Book for auto-typing. Mail offers Rules to be set and also has basic Junk Mail filtering. Multiple accounts can exist within one user's Mail configuration. It has POP3 and IMAP functionality and can retrieve Hotmail, Gmail, and .Mac email.

.Mac and Related Evidence

.Mac

.Mac is an internet resource available from Apple Inc. Features include email (5 possible addresses), web site hosting, and iDisk storage of files. This service is subscribed to on a yearly basis. A user may store files here, Backup files, Address Book entries, Safari bookmarks, Quicken data, etc. Any application that supports iDisk will be a potential area of evidence. Information can be automatically synced from a Macintosh to the iDisk, and multiple Macintosh can be configured to sync with this iDisk. Below is a screen capture of the plist file showing Moof's House is set to automatically sync with the associated iDisk.



.Mac plist Window

Safari, and Other Web Browsers

Safari

Safari is the bundled web browser with all versions of MacOS X. The browser is the most predominantly used browser, but certainly not the only one. Safari offers excellent History and Cache remnants in it's default configuration.

Other web browsers that may be installed include Mozilla, Netscape, Firefox, Opera, and Internet Explorer. There are others. Look in the Applications folder to see what has been installed and then looked for the associated setup files, bookmarks, and history in the users' Library folder.

iChat, and Instant Messaging Applications

iChat

iChat is the bundled instant messaging client in MacOS X. As of version 10.3, iChat became known as iChat AV because of the added video capability. iChat uses .Mac accounts as well as AOL Instant Messenger screen names natively. iChat also will interface with any instant messaging technology that uses "Jabber". An added feature for .Mac members is the ability to encrypt the iChat conversations. This only occurs between two .Mac members.

Other chat applications include AOL Instant Messenger, Adium, Microsoft Messenger, Skype, and SMS based applications or widgets. Look in the Applications folder to see what has been installed and then looked for the associated setup files users' Library folder or Home folder.

Mac OS X Log Files

Mac OS X, like Linux and other UNIX variants, keeps many log files. Some of the files are very detailed, yet of little use forensically. Other logs, seemingly innocuous, contain direct or indirect evidence to a users actions and intentions. Some log files will directly state exactly what a user was doing and the log entry itself would show the crime. Other entries will be indirect, yet help establish the circumstantial evidence of the crime committed. The Console utility, typically found in the /Applications/Utilities folder is where most logs can be read natively. Here are some, but certainly not all of the log files than can help establish time-tables, actions, and configurations.

Log File	Uses
/var/log/crashreporter.log	Application Usage History, information is written here when an application crashes only.
/var/log/cups/access_log	Printer Connection Information
/var/log/cups/error_log	Printer Connection Information
/var/log/daily.out	Network Interface History
/var/log/samba/log.nmbd	Samba (Windows based machine) connection information
~/Library/Logs	Any logs in this area will be specific to the user of this Home directory. Application specific logs will be found here
~/Library/Logs/DiscRecording.log	Log of CD or DVD media burned using the Finder. This is specific to the user of this Home directory.
~/Library/Logs/DiskUtility.log	Log of CD or DVD media burned using the Finder, mount and unmount history of ISO or DMG image files, Permission Repair history. and hard disk partition information.
~/Library/Logs/ iChatConnectionErrors	Log files here contain information of past iChat connection attempts. Data such as username, IP address, and Date&Time of the attempt
~/Library/Logs/Sync	Log files here will contain information on .Mac syncing, mobile devices such as iPods and cell phones, and Date&Time of the activities

Mac OS X “plist” Files

Mac OS X, and all versions of the Macintosh operating systems, do not use a registry like Microsoft Windows. User settings are “remembered” through the use of “plist” files. Plist stands for Property List Format file. There is a MAN page describing the file in detail. Here is an excerpt from the Description:

> Property lists organize data into named values and lists of values using several Core Foundation types: CFString, CFNumber, CFBoolean, CFDate, CFData, CFArray, and CFDictionary. These types give you the means to produce data that is meaningfully structured, transportable, storable, and accessible, but still as efficient as possible. The property list programming interface allows you to convert hierarchically structured combinations of these basic types to and from standard XML. The XML data can be saved to disk and later used to reconstruct the original Core Foundation objects. Note that property lists should be used for data that consists primarily of strings and numbers because they are very inefficient when used with large blocks of binary data.

This description shows us that the data is more complex than a simple “Cookie” and not easily read with a standard text editor. A Utility from Apple called “Property List Editor” will reveal the data contained within each of these files in a user friendly way. As implied by the title, it will also allow you to edit the content, so be very careful! The utility is part of the Developer tools XCode, freely available from Apple Inc. The following table lists some, but certainly not all of the valuable plist files. You will find application specific plist files created, and they will always be worth looking at for forensic data.

In the event you haven’t downloaded the XCode tools, it is still possible to look a a plist file. The plist file is likely stored in binary XML format. Opening this type of file in TextEdit will yield nothing useful. Fortunately, the Terminal command plutil converts plist file to XML format. The MAN entry for plutil is as follows:

```
NAME
     plutil -- property list utility

SYNOPSIS
     plutil [command_option] [other_options] file
             ...

DESCRIPTION
     plutil can be used to check the syntax of property list files, or convert
     a plist file from one format to another.
```

Be certain that your destination file is saved on YOUR drive and not a target drive.

The following list contains miscellaneous files, their location, and use.

File	Uses
/System/Library/CoreServices/SystemVersion.plist	Contains the current version of the installed operating system
/private/var/log/OSInstall.custom	Contains the date and time the operating system was first installed (completion time, not start time)
/private/etc/hosts	Contains defined IP addresses and the associated name

The following PLIST files can be found in the user home directory ~/Library/Preferences/

File	Uses
AddressBookMe.plist	Contains the data this user has entered about him/her self
com.apple.Bluetooth.plist	Contains devices that have connected via Bluetooth. It will show last connection date as well.
com.apple.dashboard.plist	Contains information on installed Widgets for this user.
com.apple.dock.plist	Contains information on applications available in the Dock
com.apple.DotMacSync.plist	Contains information on items to be synced as well as how often the sync is done
com.apple.finder.plist	Contains information on Recently opened folders, last server connection from Finder and the last "Go to Folder" selection
com.apple.Grab.plist	Last directory a capture was saved.
com.apple.iChat.AIM.plist	AOL Instant Messenger information
com.apple.iChat.Jabber.plist	Jabber account information
com.apple.mail.plist	Information on Mail.app setup including account names and where the email is stored locally
com.apple.NetworkUtility.plist	Information on network lookups such as Lookups, Whois, Ping and Port Scans.
com.apple.Preview.bookmarks.plist	Recent Documents opened using Preview.app
com.apple.print.PrintCenter.plist	Information on recently connected to printers
com.apple.quicktimeplayer.plist	Recently viewed movie files
com.apple.Safari.plist	History from the web browser Safari, including Recent Search terms, Recent folders utilized locally
com.scheduler.plist	Scheduled activities to run automatically such a .Mac sync or Software Update
com.apple.sidebarlists.plist	Contains a History or Current and Past item that have shown up in the Finder Windows Sidebar. It will show System assigned items as well as the items in the Custom portion of the window.

File	Uses
com.apple.systemuiserver.plist	Contains a list of the custom “menus” installed by the user. Useful in showing what runs on the machine when a user logs in.
com.RealNetworks.RealPlayer.plist	Recent audio and video clips

Again, this table is by no means complete. Using the Property List Editor, view each and any PLIST file that seems to be relevant. Many times, when software changes in version, a new PLIST file is used.

Sleep and Safe Sleep

/private/var/vm/sleepimage - This file is on Intel Macintosh portable computers to save contents of RAM to the hard disk. Its use is to recover from a power outage *during* sleep mode or when the battery is just about to run out of power during use. As of this writing, the file is written to disk, unencrypted, and yields many usual artifacts of user history, inclusive of passwords. All Macintoshes running OS X can go into sleep mode, but the computer must support "safe sleep" (sometimes referred to as Deep Sleep) to have this functionality. It is possible to turn off the safe sleep function from the command line, but not thru the System Preferences.

Detailed Macintosh Techniques

First off, the Macintosh has many, many key combinations that cause different actions right from the initial power on. Not every key combo works on every Macintosh. Most work on most Macs. That is the best that can be said. Document which ones you try for the specific case at hand, and also for future reference.

Apple Boot Key Combos

Function	Key Combination
Bypass startup drive and boot from external (or CD)	CMD-OPT-SHIFT-DELETE
Boot from CD	C
Boot from a specific SCSI ID #	CMD-OPT-SHIFT-DELETE-#
Eject Floppy Disk	Hold down Mouse button
Select Volume to start from	OPT
Start in Target Disk Mode	T
OS X Verbose Boot	CMD-V
OS X Single User Mode	CMD-S
Open Firmware	CMD-OPT-O-F

Create a Brute Force Dictionary File

The MacOS X Terminal makes it rather easy to create a brute force dictionary for attacking various encoded files. It certainly isn't a guarantee, but it offers hope. Creating this dictionary is useful when the source is not encrypted. For instance, if you try to make a dictionary file from a sparseimage file, you will get nothing useful. However, making a dictionary from the entire device may yield the password to a user's login, a website, their keychain, and so-on.

The terminal command "strings" can create a text file with the useful words contained in a file or raw device. The MAN entry for "strings" is as follows:

```
strings - find the printable strings in a object, or other binary, file
```

We can use this against a device file such as /dev/disk0 or against an unencrypted DMG file such as /Evidence/sample.dmg and have a text file created with the useful strings. The command would look like this:

```
Moofs-House:~ moof$ strings /Evidence/UnencryptedDMG.dmg > /Evidence/strings.txt
```

This command will output a text file that contains all of the useful strings contained in the DMG file. You can now use this file as a "dictionary" in a brute force attack on passwords. It might be further useful to take the repeated strings out of this file.

Useful Artifacts and Commands

As with any operating system or file system, there are numerous places to look for evidence. The Macintosh is no exception. The following tables begin to list areas of interest.

Table 1 - Artifacts

Artifact	Location
Internet History	Safari = /Users/<username>/Libary/Safari/History.plist (dates are in Absolute Date Format) Note: if the file /Users/<username>/Library/Preferences/com.Apple.Safari.plist contains the value "WebKitPrivateBrowsingEnabled" set to TRUE, no browsing history will be kept. Internet Explorer = /Users/<username>/Library/Preferences/Explorer/History.html
Email	Perform a search for files with the following extensions: .mbx, .mbox, .emlx, .imapmbox, .eml, .msf Microsoft Entourage uses a file named "database".
iPod	Perform a search for the file "com.Apple.iPod.plist". It will contain information such as serial number of the iPod, last connect time, use count, etc.
Limewire	limewire.props contains last used forward facing IP address
IP Address Info	IP Address info may be found in any of the following locations: /var/log/ipfw /var/log/secure /var/log/system I also suggest looking at other logs kept in this directory!

Table II - Terminal Window Commands

Command Line	Function
ls -al |more	"ls" is the command to list the directory contents (Present Working Directory). Adding the "-al" switch will give all entries including hidden files and show "long" entries. "Long" entries simply means you will see the associated information for each entry, rather than just the name. The "| more" is the pipe command to send the output to the "more" command. "more" is a command that will list the screen output one page at a time, pausing every 24 lines. This causes the directory listing to pause, rather than just go flying by. Some people prefer the "less" command. Read the MAN pages and choose for yourself.
pwd (Present Working Directory)	This will simply out the path of your current directory. Sitting at a "$" prompt isn't always the most useful and its easy to get lost when navigating the disk hierarchy.
find / -name "*.jpg" -print	This command will list all files, path included, that match the expression *.jpg starting from the root of the file structure. This is an example of crude searching for possible image files. Change the starting location for the search by changing the "/" to the path of choice. An example might be /Users/<username> where <username> is a valid home directory.
date -u	Displays the current system date and time in GMT

References

Information in this document has been gathered from years of education, training, and work experience. I would also be remiss if I did not mention training, websites and mailing lists that I read often, with great respect.

Many thanks go to the resources of:

- Apple Inc. including the Support and Developer websites. The information on these websites is an Examiner's greatest tool to understanding any analysis.
- Blackbag Technologies training courses
- Derrick Donnelly's email list "macos_forensics@yahoogroups.com"
- Apple Inc. Forensic email Listserv (Government email participants only at this time)
- Guidance Software discussion forums and their technical support personnel

Websites

http://www.macintouch.com

http://www.macfixit.com

http://www.apple.com/support

http://developer.apple.com

http://www.macnn.com/headlines

http://guide.apple.com

http://www.blackbagtech.com

http://www.macforensicslab.com

http://www.macosxhints.com

http://www.ifixit.com/Guide/

http://www.guidancesoftware.com

http://www.accessdata.com

Recommended Utilities and Applications

Apple Inc.

- XCode
- Property List Editor

Weird Kid Software Products

- Emailchemy

SubrosaSoft Inc.

- MacForensicLab
- DasBoot

BlackBag Technologies Inc.

- Forensic Suite

Ian Page

- MacTracker (www.mactracker.ca)

Many, MANY others as your cases develop. Use your favorite search engine, or try:

http://www.macupdate.com

http://www.versiontracker.com/macosx

MacOS X 10.4 Command Line Utilities and Daemons

```
apropos              - search the whatis database for strings
arp                  - address resolution display and control
asr                  - Apple Software Restore; copy volumes (e.g. from disk images)
atlookup             - looks up network-visible entities (NVEs) registered on the
AppleTalk network system
autodiskmount        - disk support tool
automount            - automatic server mount / unmount daemon
awk                  - pattern-directed scanning and processing language
basename, dirname    - return filename or directory portion of pathname
bash                 - GNU Bourne-Again Shell
bless                - set volume bootability and startup disk options
blued                - The Mac OS X bluetooth daemon
bootparamd           - boot parameter server
bzcmp, bzdiff        - compare bzip2 compressed files
bzgrep, bzfgrep, bzegrep - search possibly bzip2 compressed files for a regular ex-
pression
bzip2, bunzip2       - a block-sorting file compressor, v1.0.2 bzcat - decompresses
files to stdout bzip2recover - recovers data from damaged bzip2 files
cal                  - displays a calendar
calendar             - reminder service
cat                  - concatenate and print files
chflags              - change file flags
chgrp                - change group
chmod                - change file modes or Access Control Lists
chown                - change file owner and group
chpass, chfn, chsh   - add or change user database information
chroot               - change root directory
cksum, sum           - display file checksums and block counts
cksum(n)             - calculate a cksum(1) compatible checksum
clear                - clear the terminal screen
cmp                  - compare two files byte by byte
compress, uncompress - compress and expand data
configd              - System Configuration Daemon
cp                   - copy files
cron                 - daemon to execute scheduled commands (Vixie Cron)
```

```
crontab              - maintain crontab files for individual users (V3)
cupsd                - common unix printing system daemon
cvs                  - Concurrent Versions System
date                 - display or set date and time
dd                   - convert and copy a file
defaults             - access the Mac OS X user defaults system
df                   - display free disk space
diff                 - compare files line by line
diff3                - compare three files line by line
diffpp               - pretty-print diff outputs with GNU enscript
diffstat             - make histogram from diff-output
dig                  - DNS lookup utility
disable, enable      - stop/start printers and classes
diskarbitrationd     - disk arbitration daemon
disklabel            - manipulate and query an Apple Label disk label
disktool             - disk support tool
diskutil             - Modify, verify and repair local disks
ditto                - copy files and directories to a destination directory
dmesg                - display the system message buffer
domainname           - set or print the name of the current NIS domain
drutil               - interact with CD/DVD burners
dscl                 - Directory Service command line utility
du                   - display disk usage statistics
dump                 - filesystem backup
dumpfs               - dump file system information
dynamic_pager        - dynamic pager external storage manager
echo                 - write arguments to the standard output
ed                   - text editor
emacs                - GNU project Emacs
enscript             - convert text files to PostScript
env                  - set and print environment
expand, unexpand     - expand tabs to spaces, and vice versa
expr                 - evaluate expression
fdisk                - DOS partition maintenance program
fibreconfig          - Tool for configuring settings for Fibre Channel controllers
and targets
file                 - determine file type
find                 - walk a file hierarchy
```

```
fsck                - filesystem consistency check and interactive repair
fsck_hfs            - HFS file system consistency check
fsck_msdos          - DOS/Windows (FAT) file system consistency check
ftp                 - Internet file transfer program
getconf             - retrieve standard configuration variables
gpt                 - GUID partition table maintenance utility
grep, egrep, fgrep  - print lines matching a pattern
groups              - show group memberships
gzexe               - compress executable files in place
gzip, gunzip, zcat  - compress or expand files
hdik                - lightweight in-kernel disk image mounting tool
hdiutil             - manipulate disk images (attach, verify, burn, etc)
head                - display first lines of a file
heap                - List all the malloc-allocated buffers in the process's heap
hexdump, hd         - ASCII, decimal, hexadecimal, octal dump
host                - DNS lookup utility
hostname            - set or print name of current host system
ifconfig            - configure network interface parameters
info                - read Info documents
installer           - system software and package installer tool
ioreg               - show I/O Kit registry
iostat              - report I/O statistics
ip6                 - Enable or disable IPv6 on active interfaces
ip6config           - Configure IPv6 and 6to4 IPv6 tunnelling
ip6fw               - controlling utility for IPv6 firewall
ipconfig            - view and control IP configuration state
ipfw                - IP firewall and traffic shaper control program
jar                 - Java archive tool
java                - Java interpreter
kadmin              - Kerberos V5 database administration program
kadmind             - KADM5 administration server
kdb5_util           - Kerberos database maintainance utility
kextload            - loads, validates, and generates symbols for a kernel exten-
sion (kext)
kextstat            - display status of dynamically loaded kernel extensions
kextunload          - terminates and unloads kernel extensions
kill                - terminate or signal a process
killall             - kill processes by name
```

ktrace - enable kernel process tracing
last - indicate last logins of users and ttys
lastcomm - show last commands executed in reverse order
launchctl - Interfaces with launchd
launchd - System wide and per-user daemon/agent manager
ldapsearch - LDAP search tool
ldapwhoami - LDAP who am i? tool
less - opposite of more
lessecho - expand metacharacters, such as * and ?, in filenames on Unix systems
ln, link - make links
locale - display locale settings
locate - find files
login - log into the computer
logname - display user's login name
logresolve - resolve hostnames for IP-adresses in Apache logfiles
look - display lines beginning with a given string
lookupd - directory information and cache daemon
ls - list directory contents
lsbom - list contents of a bom file
lsof - list open files
lsvfs - list known virtual file systems
machine - print machine type
man - format and display the on-line manual pages
md5 - calculate a message-digest fingerprint (checksum) for a file
mdfind - finds files matching a given query
megaraid - Command Line Utility for MegaRAID management
merge - three-way file merge
mesg - display (do not display) messages from other users
mkdir - make directories
mnthome - mount an AFP (AppleShare) home directory with the correct privileges
mount - mount file systems
mount.cifs - mount using the Common Internet File System (CIFS)
mount_afp - mount an afp (AppleShare) filesystem
mount_cd9660 - mount an ISO-9660 filesystem
mount_cddafs - mount an Audio CD
mount_fdesc - mount the file-descriptor file system
mount_ftp - mount a FTP filesystem

```
mount_hfs            - mount an HFS/HFS+ file system
mount_msdos          - mount an MS-DOS file system
mount_nfs            - mount NFS file systems
mount_ntfs           - mount an NTFS file system
mount_smbfs          - mount a shared resource from an SMB file server
mount_udf            - mount a UDF filesystem
mount_webdav         - mount a WebDAV filesystem
mountd               - service remote NFS mount requests
msgs                 - system messages and junk mail program
mtree                - map a directory hierarchy
mv                   - move files
named                - Internet domain name server
nano                 - Nano's ANOther editor, an enhanced free Pico clone
natd                 - Network Address Translation daemon
net                  - Tool for administration of Samba and remote CIFS servers
netinfod             - NetInfo daemon
netstat              - show network status
newfs                - construct a new file system
newfs_hfs            - construct a new HFS Plus file system
newfs_msdos          - construct a new MS-DOS (FAT) file system
nfsd                 - remote NFS server
nice                 - execute a utility with an altered scheduling priority
nologin              - politely refuse a login
notifyd              - notification server
ntpd                 - Network Time Protocol (NTP) daemon
ntpdate              - set the date and time via NTP
ntptrace             - trace a chain of NTP servers back to the primary source
nvram                - manipulate Open Firmware NVRAM variables
open                 - open files and directories
open-x11             - run X11 programs
pagesize             - print system page size
passwd               - modify a user's password
paste                - merge corresponding or subsequent lines of files
patch                - apply a diff file to an original
pbcopy, pbpaste      - provide copying and pasting to the pasteboard (the Clip-
board) from command line
pcscd                - PC/SC Smartcard Daemon
pdisk                - Apple partition table editor
```

```
ping                - send ICMP ECHO_REQUEST packets to network hosts
ping6               - send ICMPv6 ECHO_REQUEST packets to network hosts
pl                  - converts between ASCII and binary plist formats
plutil              - property list utility
pmset               - modify power management settings
portmap             - RPC program,version to DARPA port mapper
pr                  - print files
printenv            - print out the environment
printf              - formatted output
ps                  - process status
pwd                 - return working directory name
quot                - display total block usage per user for a file system
quota               - display disk usage and limits
quotacheck          - filesystem quota consistency checker
quotaon, quotaoff   - turn filesystem quotas on and off
rarpd               - Reverse ARP Daemon
rcp                 - remote file copy
reboot, halt        - stopping and restarting the system
renice              - alter priority of running processes
repquota            - summarize quotas for a file system
restore             - restore files or file systems from backups made with dump
rev                 - reverse lines of a file
rlogin              - remote login
rm, unlink          - remove directory entries
rmdir               - remove directories
routed              - network RIP and router discovery routing daemon
rsh                 - remote shell
rwho                - who is logged in on local machines
rwhod               - system status server
say                 - Convert text to audible speech
scp                 - secure copy (remote file copy program)
screencapture       - capture and manipulate clipboard contents
screenreaderd       - VoiceOver daemon
sftp                - secure file transfer program
sftp-server         - SFTP server subsystem
showmount           - show remote nfs mounts on host
shutdown            - close down the system at a given time
sleep               - suspend execution for an interval of time
```

```
smbclient           - ftp-like client to access SMB/CIFS resources on servers
smbd                - server to provide SMB/CIFS services to clients
smbstatus           - report on current Samba connections
snmpd               - daemon to respond to SNMP request packets
snmptable           - retrieve an SNMP table and display it in tabular form
snmptrapd           - Receive and log SNMP trap messages
sort                - sort lines of text files
split               - split a file into pieces
spray               - send many packets to host
srm                 - securely remove files or directories
ssh                 - OpenSSH SSH client (remote login program)
sshd                - OpenSSH SSH daemon
stat, readlink      - display file status
strings             - find the printable strings in a object, or other binary,
file
strip               - remove symbols
su                  - substitute user identity
sudo, sudoedit      - execute a command as another user
sum(n)              - calculate a sum(1) compatible checksum
sw_vers             - print Mac OS X operating system version information
sync                - force completion of pending disk writes (flush cache)
syslog              - Apple System Log utility
syslog.conf(5)      - syslogd(8) configuration file
syslogd             - Apple System Log server
system_profiler     - reports system hardware and software configuration
tail                - display the last part of a file
talk                - talk to another user
tar                 - tape archiver; manipulate "tar" archive files
tcpdump             - dump traffic on a network
tcsh                - C shell with file name completion and command line editing
telnet              - user interface to the TELNET protocol
tftp                - trivial file transfer program
tim                 - authetication server
time                - time command execution
timed               - time server daemon
timutil             - authetication server utility
top                 - display and update sorted information about processes
touch               - change file access and modification times
```

```
traceroute            - print the route packets take to network host
traceroute6           - print the route IPv6 packets will take to the destination
tty                   - return user's terminal name
umount                - unmount filesystems
uname                 - Print operating system name
uniq                  - report or filter out repeated lines in a file
unzip                 - list, test and extract compressed files in a ZIP archive
update                - flush internal filesystem caches to disk frequently
update_prebinding     - Update prebinding information when new system libraries or
frameworks are installed
uptime                - show how long system has been running
users                 - list current users
uuencode, uudecode    - encode/decode a binary file
vers_string           - produce version identification string
vim                   - Vi IMproved, a programmers text editor
vipw                  - edit the password file
visudo                - edit the sudoers file
vpnd                  - Mac OS X VPN service daemon
w                     - display who is logged in and what they are doing
wc                    - word, line, character, and byte count
whatis                - search the whatis database for complete words
whereis               - locate programs
which                 - locate a program file in the user's path
who                   - display who is on the system
whoami                - display effective user id
whois                 - Internet domain name and network number directory service
winbindd              - Name Service Switch daemon for resolving names from NT serv-
ers
write                 - send a message to another user
xgrid                 - submit and monitor xgrid jobs
xinetd                - the extended Internet services daemon
zcmp, zdiff           - compare compressed files
zgrep                 - search possibly compressed files for a regular expression
zip, zipcloak, zipnote, zipsplit - package and compress (archive) files
zipgrep               - search files in a ZIP archive for lines matching a pattern
zipinfo               - list detailed information about a ZIP archive
zsh                   - the Z shell
```

never give up...the answer is right in front of you